Exact Subpixel Motion Estimation in DCT Domain

by U-V. Koc and K.J.R. Liu

T.R. 96-2

ISR
INSTITUTE FOR SYSTEMS RESEARCH

*Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry*

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **1996**	2. REPORT TYPE	3. DATES COVERED **00-00-1996 to 00-00-1996**

4. TITLE AND SUBTITLE **Exact Subpixel Motion Estimation in DCT Domain**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical Engineering,Institute for Systems Research,University of Maryland,College Park,MD,20742**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **32**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Exact Subpixel Motion Estimation In DCT Domain

Ut-Va Koc and *K. J. Ray Liu*

Electrical Engineering Department and Institute for Systems Research
University of Maryland at College Park
College Park, Maryland 20742
koc@eng.umd.edu and kjrliu@eng.umd.edu

ABSTRACT

Currently existing subpixel motion estimation algorithms require interpolation of inter-pixel values which undesirably increases the overall complexity and data flow and deteriorates estimation accuracy. In this paper, we develop DCT-based techniques to estimate subpel motion at different desired subpel levels of accuracy in DCT domain without interpolation. We show that subpixel motion information is preserved in the DCT of a shifted signal under some condition in the form of pseudo phases and establish subpel sinusoidal orthogonal principles to extract this information. Though applicable to other areas as well, the resulted algorithms from these techniques for video coding are flexible and scalable in terms of estimation accuracy with very low computational complexity $O(N^2)$ compared to $O(N^4)$ for Full Search Block Matching Approach and its subpixel versions. Above all, motion estimation in DCT domain instead of spatial domain simplifies the conventional hybrid DCT-based video coder, especially the heavily loaded feedback loop in the conventional design, resulting in a fully DCT-based high-throughput video codec. In addition, the computation of pseudo phases is local and thus a highly parallel architecture is feasible for the DCT-based algorithms. Finally simulation on video sequences of different characteristics shows comparable performance of the proposed algorithms to block matching approaches.

Keywords: motion estimation, subpixel accuracy, video coding, video compression, shift measurement

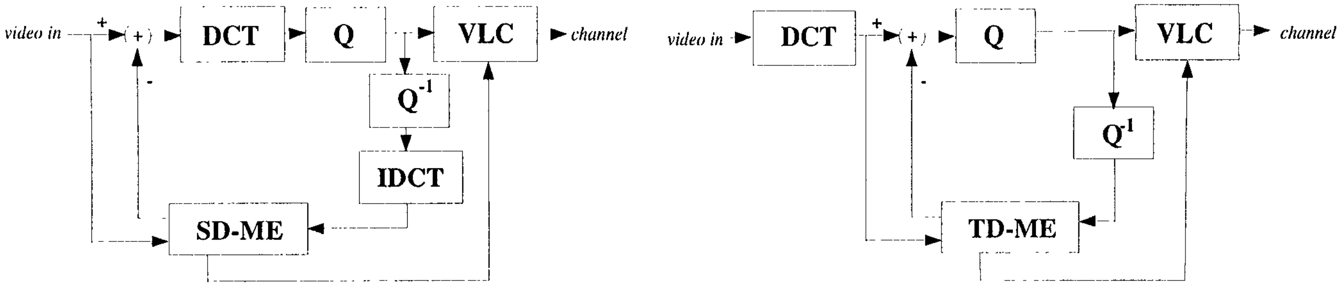
This work is supported in part by the ONR grant N00014-93-1-0566, NSF grant MIP9457397, and MIPS/MicroStar.

I. Introduction

Accurate estimation of displacement or location of a signal or image is important in many applications of signal and image processing such as time delay estimation [21], target tracking [35], non-contact measurement [40], [2], remote sensing [4], [11], computer vision [1], image registration [8], [38], and so on. In video coding, motion estimation is proved to be very useful for reduction of temporal redundancy. Therefore, a number of motion estimation algorithms have been devised solely for video coding [29], [10] and numerous VLSI architectures have been designed for practical video applications [33]. To further improve the compression rate, motion estimation with subpixel accuracy is essential because movements in a video sequence are not necessarily multiples of the sampling grid distance in the rectangular sampling grid of a camera. It is shown that significant improvement of coding gain can be obtained with motion estimation of half pixel or finer accuracy [16]. Further investigation reveals that the temporal prediction error variance is generally decreased by subpixel motion compensation but beyond a certain "critical accuracy" the possibility of further improving prediction by more accurate motion compensation is small [13]. As suggested in [16], [12], motion compensation with 1/4-pel accuracy is sufficiently accurate for broadcast TV signals, but for videophone signals, half-pel accuracy is good enough. As a result, motion compensation with half-pel accuracy is recommended in MPEG standards [27], [28]. Implementations of half-pel motion estimation have started to be realized [39], [3], [6].

Many subpixel motion estimation schemes have been proposed over the years [1], [29], [10]. The most commonly used spatial-domain fractional-pel motion estimation algorithms such as the block matching approach [26], [12], [9], the pel-recursive approach [30], [31], require interpolation of images through bilinear, Lagrange, or other interpolation methods [34]. However, interpolation not only increases the complexity and data flow of a coder but also may adversely affect the accuracy of motion estimates from the interpolated images [12]. It is more desirable that subpixel accuracy of motion estimates can be obtained without interpolating the images. In the category of frequency-domain methods, the phase correlation technique [37], [41], [22] is reported to provide accurate estimates without inter-pixel interpolation but is based on Fast Fourier Transform (FFT), which is incompatible with DCT-based video coding standards and requires a large search window at a high computational cost. Other FFT-based approaches such as in [17], [20] also have similar drawbacks.

Due to the fact that the motion compensated DCT-based hybrid approach is the backbone of several



(a) Hybrid Coder

(b) Fully DCT-Based Coder

Fig. 1. Coder structures: (a) Commonly used motion-compensated DCT hybrid coder performs motion estimation in the spatial domain. (b) Fully DCT-based coder estimates motion in the transform domain.

international video coding standards such as CCITT H.261 [14], MPEG1 [27], MPEG2 [28], and the emerging HDTV [5] and H.263 [15] standards. it is more desirable to estimate motion with fractional-pel accuracy *without* any inter-pixel interpolation at a low computational cost in the DCT domain so that seamless integration of the motion compensation unit with the spatial compression unit is possible. More specifically, a conventional standard-compliant video coder is usually implemented as a hybrid DCT-based structure in Fig. 1(a), which achieves spatial compression through Discrete Cosine Transform (DCT) and temporal compression through motion compensation traditionally accomplished in spatial domain. In this hybrid structure, the feedback loop contains three major components: DCT, IDCT (Inverse DCT) and SD-ME (spatial domain motion estimation). All incoming raw video data must traverse this heavily loaded feedback loop once in order to be encoded in the output bitstream. In addition to the disadvantage of having more hardware components, the throughput of the whole coder is also limited by the complexity of the loop. However, if motion can be estimated and compensated entirely in the transform domain, then DCT can be moved out of the loop and IDCT be eliminated, resulting in a fully DCT-based video coder as shown in Fig. 1 (b) where the feedback loop has only one major component, transform domain motion estimation (TD-ME) [23] instead of three major components.

Based upon the concept of pseudo-phases in DCT coefficients and the sinusoidal orthogonal principles, a DCT-based integer-pel motion estimation scheme (DXT-ME) of very low computational complexity ($O(N^2)$ as opposed to $O(N^4)$ for the widely used Full Search Block Matching Algorithm) was proposed in [18], [19] to realize the fully DCT-based video coder design, as depicted in Fig. 2 and summarized in Table I. In this paper, we further explore this DCT-based concept at the subpixel level and show that if the spatial sampling of images satisfies the Nyquist criterion, the subpixel motion information is

1. Compute the 2-D DCT coefficients of second kind (2D-DCT-II) of a $N \times N$ block of pixels at the current frame t, $\{x_t(m,n); m,n \in \{0,...,N-1\}\}$.
2. Convert the stored 2D-DCT-II coefficients of the corresponding $N \times N$ block of pixels at the previous frame $t-1$, $\{x_{t-1}(m,n); m,n \in \{0,...,N-1\}\}$ to 2D DCT coefficients of first kind (2D-DCT-I) through a simple rotation unit T.
3. Find the pseudo phases $\{g^{CS}(k,l);\ k = 0,1,\ldots,N-1;\ l = 1,2,\ldots,N\}$ and $\{g^{SC}(k,l);\ k = 1,2,\ldots,N;\ l = 0,1,\ldots,N-1\}$, which are calculated from the DCT coefficients independently at each spectral location (k,l).
4. Determine the normalized pseudo phases $f(k,l)$ and $g(k,l)$ from $g^{CS}(k,l)$ and $g^{SC}(k,l)$ respectively by setting ill-formed $g^{CS}(k,l)$ and $g^{SC}(k,l)$ to zero:

$$f(k,l) = \begin{cases} C(k)C(l)g^{CS}(k,l), & \text{for } |g^{CS}(k,l)| \le 1, \\ 0, & \text{otherwise,} \end{cases}$$
$$g(k,l) = \begin{cases} C(k)C(l)g^{SC}(k,l), & \text{for } |g^{SC}(k,l)| \le 1, \\ 0, & \text{otherwise,} \end{cases}$$

   where

$$C(n) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } n = 0 \text{ or } N, \\ 1, & \text{otherwise,} \end{cases}$$

5. Obtain the inverse DCT (2D-IDCT-II) of $f(k,l)$ and $g(k,l)$ as $DCS(m,n)$ and $DSC(m,n)$ for $m,n \in \{0,...,N-1\}$ respectively which basically are composed of impulse functions whose peak positions indicate the shift amount and peak signs reveal the direction of the movement:

$$DCS(m,n) = \frac{4}{N^2}\sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l)\cos\frac{k\pi}{N}(m+\frac{1}{2})\sin\frac{l\pi}{N}(n+\frac{1}{2}),$$
$$DSC(m,n) = \frac{4}{N^2}\sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l)\sin\frac{k\pi}{N}(m+\frac{1}{2})\cos\frac{l\pi}{N}(n+\frac{1}{2}).$$

6. Search for the peaks of $DCS(m,n)$ and $DSC(m,n)$ over $(m,n) \in \{0,\ldots,N-1\}^2$ (or range of interest) such that

$$(i_{DCS}, j_{DCS}) = arg \max_{m,n \in \Phi} |DCS(m,n)|,$$
$$(i_{DSC}, j_{DSC}) = arg \max_{m,n \in \Phi} |DSC(m,n)|.$$

7. Estimate the displacement $\hat{d} = (\hat{m}_u, \hat{m}_v)$ from the signs and positions of the peaks of $DCS(m,n)$ and $DSC(m,n)$:

$$\hat{m}_u = \begin{cases} i_{DSC} = i_{DCS}, & \text{if } DSC(i_{DSC}, j_{DSC}) > 0, \\ -(i_{DSC}+1) = -(i_{DCS}+1), & \text{if } DSC(i_{DSC}, j_{DSC}) < 0, \end{cases}$$
$$\hat{m}_v = \begin{cases} j_{DCS} = j_{DSC}, & \text{if } DCS(i_{DCS}, j_{DCS}) > 0, \\ -(j_{DCS}+1) = -(j_{DSC}+1), & \text{if } DCS(i_{DCS}, j_{DCS}) < 0, \end{cases}$$

TABLE I

Summary of DCT-based integer-pel motion estimation scheme (DXT-ME)

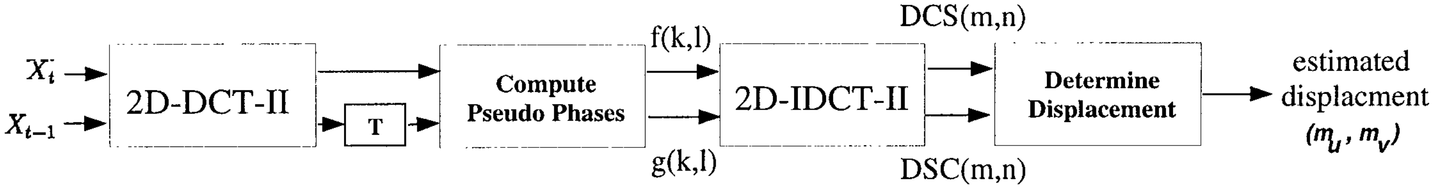


Fig. 2. Block diagram of the DCT-based integer-pel motion estimator (DXT-ME)

preserved in the pseudo phases of DCT coefficients of moving images. Furthermore it can be shown that with appropriate modification, the sinusoidal orthogonal principles can still be applicable except that an impulse function is replaced by a sinc function whose peak position reveals subpixel displacement. Therefore, exact subpixel motion displacement can be obtained without the use of interpolation. From these observations, we can develop a set of subpixel DCT-based motion estimation algorithms, that are fully compatible with the integer-pel motion estimator, for low-complexity and high-throughput video applications.

In this paper, we discuss the pseudo phases carrying subpixel motion information in Section II and the subpel sinusoidal orthogonal principles in Section III for objects moving out of synchronization with the sampling grid. In Section IV, we propose the DCT-based half-pel (HDXT-ME) and quarter-pel (QDXT-ME and Q4DXT-ME) motion estimation algorithms whose simulation results on actual video sequences of different characteristics are presented in Section V in comparison with the popular block matching approaches. Finally, we conclude the major contributions of this paper in Section VI.

II. Pseudo Phases at Subpixel Level

A. One Dimensional Signal Model

Without loss of generality, let us consider the one dimensional model in which a continuous signal $x_c(t)$ and its shifted version $x_c(t-d)$ are sampled at a sampling frequency $1/T$ to generate two sample sequences $\{x_1(n) = x_c(nT)\}$ and $\{x_2(n) = x_c(nT-d)\}$, respectively. Let us define the DCT and DST coefficients as

$$X_i^C(k) \triangleq \mathrm{DCT}\{x_i\} = \frac{2C(k)}{N}\sum_{n=0}^{N-1} x_i(n)\cos\frac{k\pi}{N}(n+\frac{1}{2}), \tag{1}$$

$$X_i^S(k) \triangleq \mathrm{DST}\{x_i\} = \frac{2C(k)}{N}\sum_{n=0}^{N-1} x_i(n)\sin\frac{k\pi}{N}(n+\frac{1}{2}), \tag{2}$$

where $$C(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k = 0 \text{ or } N, \\ 1, & \text{otherwise,} \end{cases}$$

for $i = 1$ or 2. By using the sinusoidal relationship:

$$\cos\frac{k\pi}{N}(n+\frac{1}{2}) = \frac{1}{2}[e^{j\frac{k\pi}{N}(n+\frac{1}{2})} + e^{-j\frac{k\pi}{N}(n+\frac{1}{2})}];\ \sin\frac{k\pi}{N}(n+\frac{1}{2}) = \frac{1}{2j}[e^{j\frac{k\pi}{N}(n+\frac{1}{2})} - e^{-j\frac{k\pi}{N}(n+\frac{1}{2})}], \tag{3}$$

we can show that the DCT/DST and DFT coefficients are related as follows:

$$X_i^C(k) = \frac{C(k)}{N}[\tilde{X}_i^Z(-k)e^{j\frac{k\pi}{2N}} + \tilde{X}_i^Z(k)e^{-j\frac{k\pi}{2N}}], \text{ for } k = 0, \ldots, N-1, \tag{4}$$

$$X_i^S(k) = \frac{C(k)}{jN}[\tilde{X}_i^Z(-k)e^{j\frac{k\pi}{2N}} - \tilde{X}_i^Z(k)e^{-j\frac{k\pi}{2N}}], \text{ for } k = 1, \ldots, N, \tag{5}$$

where $\{\tilde{X}_i^Z(k)\}$ is the DFT of the zero-padded sequence $\{x_i^Z(n)\}$ defined as

$$x_i^Z(n) = \begin{cases} x_i(n), & \text{for } n = 0, \ldots, N-1, \\ 0, & \text{for } n = N, \ldots, 2N-1, \end{cases} \tag{6}$$

so that

$$\tilde{X}_i^Z(k) \triangleq \text{DFT}\{x_i^Z\} = \sum_{n=0}^{N-1} x_i(n)e^{-j\frac{2k\pi n}{2N}}, \text{ for } k = 0, \ldots, 2N-1. \tag{7}$$

From the sampling theorem, we know that the Discrete Time Fourier Transform (DTFT) of sequences $x_1(n)$ and $x_2(n)$ are related to the Fourier Transform (FT) of $x_c(t)$, $X_c(\Omega)$, in the following way:

$$X_1(\omega) \triangleq \text{DTFT}\{x_1\} = \frac{1}{T}\sum_l X_c(\frac{\omega - 2\pi l}{T}), \tag{8}$$

$$X_2(\omega) \triangleq \text{DTFT}\{x_2\} = \frac{1}{T}\sum_l X_c(\frac{\omega - 2\pi l}{T})\ e^{-j(\frac{\omega-2\pi l}{T})d}. \tag{9}$$

Furthermore, if $X_c(\Omega)$ is bandlimited in the baseband $(-\frac{\pi}{T}, \frac{\pi}{T})$, then for $\Omega = \frac{\omega}{T} \in (-\frac{\pi}{T}, \frac{\pi}{T})$,

$$X_1(\Omega T) = \frac{1}{T}X_c(\Omega), \tag{10}$$

$$X_2(\Omega T) = \frac{1}{T}X_c(\Omega)\ e^{-j\Omega d}. \tag{11}$$

Thus, the DFT of $x_1(n)$ and $x_2(n)$ are

$$\tilde{X}_1(k) \triangleq \text{DFT}\{x_1\} = \sum_{n=0}^{N-1} x_1(n)e^{-j\frac{2\pi kn}{N}} = X_1(\frac{2\pi k}{N}) = \frac{1}{T}X_c(\frac{2\pi k}{NT}), \tag{12}$$

$$\tilde{X}_2(k) \triangleq \text{DFT}\{x_2\} = \sum_{n=0}^{N-1} x_2(n)e^{-j\frac{2\pi kn}{N}} = X_2(\frac{2\pi k}{N}) = \frac{1}{T}X_c(\frac{2\pi k}{NT})\ e^{-j\frac{2\pi kd}{NT}}, \tag{13}$$

whereas the DFT of $x_1^Z(n)$ and $x_2^Z(n)$ become

$$\tilde{X}_1^Z(k) = X_1(\frac{\pi k}{N}) = \frac{1}{T} X_c(\frac{\pi k}{NT}), \tag{14}$$

$$\tilde{X}_2^Z(k) = X_2(\frac{\pi k}{N}) = \frac{1}{T} X_c(\frac{\pi k}{NT})\, e^{-j\frac{\pi k d}{NT}}. \tag{15}$$

Therefore,

$$X_2(\frac{\pi k}{N}) = X_1(\frac{\pi k}{N}) e^{-j\frac{\pi k d}{NT}}. \tag{16}$$

Substituting (16) back into (4)-(5), we get

$$X_2^C(k) = \frac{C(k)}{N}[\tilde{X}_1^Z(-k) e^{j\frac{k\pi d}{NT}} e^{j\frac{k\pi}{2N}} + \tilde{X}_1^Z(k) e^{-j\frac{k\pi d}{NT}} e^{-j\frac{k\pi}{2N}}], \text{ for } k = 0, \ldots, N-1, \tag{17}$$

$$X_2^S(k) = \frac{C(k)}{jN}[\tilde{X}_1^Z(-k) e^{j\frac{k\pi d}{NT}} e^{j\frac{k\pi}{2N}} - \tilde{X}_1^Z(k) e^{-j\frac{k\pi d}{NT}} e^{-j\frac{k\pi}{2N}}], \text{ for } k = 1, \ldots, N. \tag{18}$$

Using the sinusoidal relationship in (3) to change natural exponents back to cosine/sine, we finally obtain the relationship between $x_1(n)$ and $x_2(n)$ in DCT/DST domain:

$$X_2^C(k) = \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_1(n) \cos \frac{k\pi}{N}(n + \frac{d}{T} + \frac{1}{2}), \text{ for } k = 0, \ldots, N-1, \tag{19}$$

$$X_2^S(k) = \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_1(n) \sin \frac{k\pi}{N}(n + \frac{d}{T} + \frac{1}{2}), \text{ for } k = 1, \ldots, N. \tag{20}$$

We conclude the result in the following theorem:

THEOREM 1: If a continuous signal $x_c(t)$ is $\frac{\pi}{T}$-bandlimited and the sampled sequences of $x_c(t)$ and $x_c(t-d)$ are $\{x_c(nT)\}$ and $\{x_c(nT-d)\}$, respectively, then their DCT and DST are related by

$$\mathrm{DCT}\{x_c(nT-d)\} = \mathrm{DCT}_{\frac{d}{T}}\{x_c(nT)\}, \tag{21}$$

$$\mathrm{DST}\{x_c(nT-d)\} = \mathrm{DST}_{\frac{d}{T}}\{x_c(nT)\}, \tag{22}$$

where

$$\mathrm{DCT}_\alpha\{x\} \triangleq \frac{2C(k)}{N} \sum_{n=0}^{N-1} x(n) \cos \frac{k\pi}{N}(n + \alpha + \frac{1}{2}), \tag{23}$$

$$\mathrm{DST}_\beta\{x\} \triangleq \frac{2C(k)}{N} \sum_{n=0}^{N-1} x(n) \sin \frac{k\pi}{N}(n + \beta + \frac{1}{2}), \tag{24}$$

are the DCT and DST with α and β shifts in their kernels, respectively. Here d is the shift amount and T is the sampling interval, but d/T is not necessarily an integer.

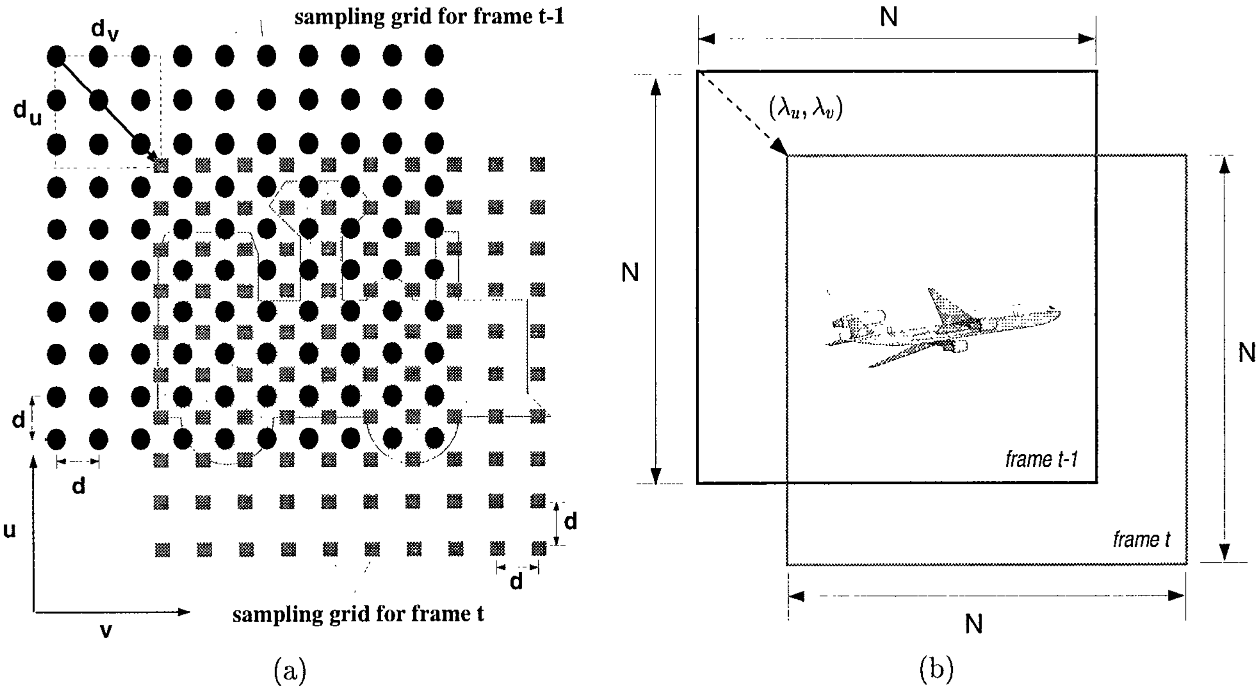


Fig. 3. (a) The black dots and the gray squares symbolize the sampling grids for frames $I_{t-1}(u,v)$ and $I_t(u,v)$ at a sampling distance d respectively. These two frames are aligned on the common object displaced by (d_u, d_v) in the continuous coordinate (u, v). (b) Two digitized images of consecutive frames, $x_{t-1}(m,n)$ and $x_t(m,n)$, are aligned on the common object moving $(\lambda_u, \lambda_v) = (d_u/d, d_v/d)$ pixels southeast.

B. Two Dimensional Image Model

Consider a moving object casting a continuous intensity profile $I_t(u,v)$ on a camera plane of the continuous coordinate (u,v) where the subscript t denotes the frame number. This intensity profile is then digitized on the fixed sampling grid of the camera with a sampling distance d to generate the current frame of pixels $x_t(m,n)$ shown in Fig. 3(a) where m and n are integers. Further assume that the displacement of the object between the frames $t-1$ and t is (d_u, d_v) such that $I_t(u,v) = I_{t-1}(u-d_u, v-d_v)$ where $d_u = (m_u + \nu_u)d = \lambda_u d$ and $d_v = (m_v + \nu_v)d = \lambda_v d$. Here m_u and m_v are the integer components of the displacement, and ν_u and $\nu_v \in (-\frac{1}{2}, \frac{1}{2}]$. Therefore,

$$x_t(m,n) = I_t(md, nd) = I_{t-1}(md - d_u, nd - d_v),$$

$$x_{t-1}(m,n) = I_{t-1}(md, nd),$$

as in Fig. 3(b). Unlike the case of integer-pel movement, the displacement is not necessarily multiples of the sampling distance d. In other words, ν_u and ν_v do not necessarily equal zero.

For integer-pel displacements, i. e. $\lambda_u = m_u$ and $\lambda_v = m_v$, the pseudo phases are computed by solving the *pseudo-phase motion equation* at (k,l) [18], [19]:

$$\mathbf{Z}_{t-1}(k,l) \cdot \vec{\theta}_{m_u,m_v}(k,l) = \vec{\mathbf{x}}_t(k,l), \text{ for } k,l \in \mathcal{N} \tag{25}$$

where $\mathcal{N} = \{1,\ldots,N-1\}$, $\vec{\theta}_{m_u,m_v}$ is the pseudo-phase vector, and the 4×4 *system matrix* $\mathbf{Z}_{t-1}$ and the vector $\vec{\mathbf{x}}_t$ are composed from the 2D-DCT-II of $x_{t-1}(m,n)$ and the 2D-DCT-I of $x_t(m,n)$ respectively:

$$\mathbf{Z}_{t-1}(k,l) = \begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) & -Z_{t-1}^{sc}(k,l) & +Z_{t-1}^{ss}(k,l) \\ Z_{t-1}^{cs}(k,l) & +Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{ss}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & -Z_{t-1}^{ss}(k,l) & +Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{ss}(k,l) & +Z_{t-1}^{sc}(k,l) & +Z_{t-1}^{cs}(k,l) & +Z_{t-1}^{cc}(k,l) \end{bmatrix},$$

$$\vec{\mathbf{x}}_t(k,l) = \begin{bmatrix} X_t^{cc}(k,l) \\ X_t^{cs}(k,l) \\ X_t^{sc}(k,l) \\ X_t^{ss}(k,l) \end{bmatrix}, \; \vec{\theta}_{m_u,m_v}(k,l) = \begin{bmatrix} g_{m_u m_v}^{CC}(k,l) \\ g_{m_u m_v}^{CS}(k,l) \\ g_{m_u m_v}^{SC}(k,l) \\ g_{m_u m_v}^{SS}(k,l) \end{bmatrix}.$$

Here the 2D-DCT-I of $x_{t-1}(m,n)$ are defined as:

$$Z_{t-1}^{cc}(k,l) = \mathrm{DCCTI}\{x_{t-1}\} \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos[\frac{k\pi}{N}(m)] \cos[\frac{l\pi}{N}(n)], \quad k,l \in \{0,\ldots,N\}, \tag{26}$$

$$Z_{t-1}^{cs}(k,l) = \mathrm{DCSTI}\{x_{t-1}\} \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos[\frac{k\pi}{N}(m)] \sin[\frac{l\pi}{N}(n)], \quad k \in \{0,\ldots,N\}, l \in \{1,\ldots,N-1\}, \tag{27}$$

$$Z_{t-1}^{sc}(k,l) = \mathrm{DSCTI}\{x_{t-1}\} \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin[\frac{k\pi}{N}(m)] \cos[\frac{l\pi}{N}(n)], \quad k \in \{1,\ldots,N-1\}, l \in \{0,\ldots,N\}, \tag{28}$$

$$Z_{t-1}^{ss}(k,l) = \mathrm{DSSTI}\{x_{t-1}\} \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin[\frac{k\pi}{N}(m)] \sin[\frac{l\pi}{N}(n)], \quad k,l \in \{1,\ldots,N-1\}, \tag{29}$$

and the 2D-DCT-II of $x_t(m,n)$ as

$$X_t^{cc}(k,l) \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cos[\frac{k\pi}{N}(m+0.5)] \cos[\frac{l\pi}{N}(n+0.5)], \quad k,l \in \{0,\ldots,N-1\}, \tag{30}$$

$$X_t^{cs}(k,l) \triangleq \frac{4}{N^2}C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cos[\frac{k\pi}{N}(m+0.5)] \sin[\frac{l\pi}{N}(n+0.5)], \tag{31}$$

$$k \in \{0, \ldots, N-1\}, l \in \{1, \ldots, N\},$$

$$X_t^{sc}(k,l) \triangleq \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \sin[\frac{k\pi}{N}(m+0.5)] \cos[\frac{l\pi}{N}(n+0.5)], \tag{32}$$

$$k \in \{1, \ldots, N\}, l \in \{0, \ldots, N-1\},$$

$$X_t^{ss}(k,l) \triangleq \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \sin[\frac{k\pi}{N}(m+0.5)] \sin[\frac{l\pi}{N}(n+0.5)], \tag{33}$$

$$k, l \in \{1, \ldots, N\},$$

where $\{Z_{t-1}^{xx}; xx = cc, cs, sc, ss\}$ can be obtained from $\{X_{t-1}^{xx}; xx = cc, cs, sc, ss\}$ by a simple rotation:

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) \\ Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{ss}(k,l) \end{bmatrix} = \begin{bmatrix} +\cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & +\cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & +\sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & \sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} \\ -\cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & +\cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} \\ -\sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & +\cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} \\ +\sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} \end{bmatrix} \begin{bmatrix} X_{t-1}^{cc}(k,l) \\ X_{t-1}^{cs}(k,l) \\ X_{t-1}^{sc}(k,l) \\ X_{t-1}^{ss}(k,l) \end{bmatrix}. \tag{34}$$

for $k, l \in \mathcal{N}$ and $\{X_{t-1}^{xx}; xx = cc, cs, sc, ss\}$ are computed and stored in memory in the previous encoding cycle.

However, for non-integer pel movement, we need to use (21)-(22) in Theorem 1 to derive the system equation at the subpixel level. If the Fourier transform of the continuous intensity profile $I_t(u,v)$ is $\frac{\pi}{d}$-bandlimited and $I_t(u,v) = I_{t-1}(u - d_u, v - d_v)$, then according to Theorem 1, we can obtain the following 2-D relations:

$$X_t^{cc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos[\frac{k\pi}{N}(m + \lambda_u + \frac{1}{2})] \cos[\frac{l\pi}{N}(n + \lambda_v + \frac{1}{2})], \tag{35}$$

$$k, l \in \{0, \ldots, N-1\},$$

$$X_t^{cs}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos[\frac{k\pi}{N}(m + \lambda_u + \frac{1}{2})] \sin[\frac{l\pi}{N}(n + \lambda_v + \frac{1}{2})], \tag{36}$$

$$k \in \{0, \ldots, N-1\}, l \in \{1, \ldots, N\},$$

$$X_t^{sc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin[\frac{k\pi}{N}(m + \lambda_u + \frac{1}{2})] \cos[\frac{l\pi}{N}(n + \lambda_v + \frac{1}{2})], \tag{37}$$

$$k \in \{1, \ldots, N\}, l \in \{0, \ldots, N-1\},$$

$$X_t^{ss}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin[\frac{k\pi}{N}(m + \lambda_u + \frac{1}{2})] \sin[\frac{l\pi}{N}(n + \lambda_v + \frac{1}{2})], \tag{38}$$

$$k, l \in \{1, \ldots, N\}.$$

Thus, we can obtain the *pseudo-phase motion equation* at the subpixel level:

$$\mathbf{Z}_{t-1}(k,l) \cdot \vec{\theta}_{\lambda_u,\lambda_v}(k,l) = \vec{\mathbf{x}}_t(k,l), \text{ for } k, l \in \mathcal{N}, \tag{39}$$

where $\vec{\theta}_{\lambda_u,\lambda_v}(k,l) = [g^{CC}_{\lambda_u,\lambda_v}(k,l), g^{CS}_{\lambda_u,\lambda_v}(k,l), g^{SC}_{\lambda_u,\lambda_v}(k,l), g^{SS}_{\lambda_u,\lambda_v}(k,l)]^T$. A similar relationship between the DCT coefficients of $x_t(m,n)$ and $x_{t-1}(m,n)$ at the block boundary can be obtained in the same way:

$$\begin{bmatrix} Z^{cc}_{t-1}(k,l) & -Z^{cs}_{t-1}(k,l) \\ Z^{cs}_{t-1}(k,l) & Z^{cc}_{t-1}(k,l) \end{bmatrix} \begin{bmatrix} g^{CC}_{\lambda_u,\lambda_v}(k,l) \\ g^{CS}_{\lambda_u,\lambda_v}(k,l) \end{bmatrix} = \begin{bmatrix} X^{cc}_t(k,l) \\ X^{cs}_t(k,l) \end{bmatrix}, \qquad k=0,\, l \in \mathcal{N}, \tag{40}$$

$$\begin{bmatrix} Z^{cc}_{t-1}(k,l) & -Z^{sc}_{t-1}(k,l) \\ Z^{sc}_{t-1}(k,l) & Z^{cc}_{t-1}(k,l) \end{bmatrix} \begin{bmatrix} g^{CC}_{\lambda_u,\lambda_v}(k,l) \\ g^{SC}_{\lambda_u,\lambda_v}(k,l) \end{bmatrix} = \begin{bmatrix} X^{cc}_t(k,l) \\ X^{sc}_t(k,l) \end{bmatrix}, \qquad l=0,\, k \in \mathcal{N}, \tag{41}$$

$$\begin{bmatrix} Z^{cc}_{t-1}(k,l) & -Z^{cs}_{t-1}(k,l) \\ Z^{cs}_{t-1}(k,l) & Z^{cc}_{t-1}(k,l) \end{bmatrix} \begin{bmatrix} g^{SC}_{\lambda_u,\lambda_v}(k,l) \\ g^{SS}_{\lambda_u,\lambda_v}(k,l) \end{bmatrix} = \begin{bmatrix} X^{sc}_t(k,l) \\ X^{ss}_t(k,l) \end{bmatrix}, \qquad k=N,\, l \in \mathcal{N}, \tag{42}$$

$$\begin{bmatrix} Z^{cc}_{t-1}(k,l) & -Z^{sc}_{t-1}(k,l) \\ Z^{sc}_{t-1}(k,l) & Z^{cc}_{t-1}(k,l) \end{bmatrix} \begin{bmatrix} g^{CS}_{\lambda_u,\lambda_v}(k,l) \\ g^{SS}_{\lambda_u,\lambda_v}(k,l) \end{bmatrix} = \begin{bmatrix} X^{cs}_t(k,l) \\ X^{ss}_t(k,l) \end{bmatrix}, \qquad l=N,\, k \in \mathcal{N}, \tag{43}$$

$$Z^{cc}_{t-1}(k,l) g^{CS}_{\lambda_u,\lambda_v}(k,l) = X^{cs}_t(k,l), \qquad k=0,\, l=N, \tag{44}$$

$$Z^{cc}_{t-1}(k,l) g^{SC}_{\lambda_u,\lambda_v}(k,l) = X^{sc}_t(k,l), \qquad k=N,\, l=0. \tag{45}$$

In (39), the pseudo phase vector $\vec{\theta}_{\lambda_u,\lambda_v}(k,l)$ contains the information of the subpixel movement (λ_u, λ_v). In an ideal situation where one rigid object is moving translationally within the block boundary without observable background and noise, we can find $\vec{\theta}_{\lambda_u,\lambda_v}(k,l)$ explicitly in terms of λ_u and λ_v as such:

$$\vec{\theta}_{\lambda_u,\lambda_v}(k,l) = \begin{bmatrix} g^{CC}_{\lambda_u,\lambda_v}(k,l) \\ g^{CS}_{\lambda_u,\lambda_v}(k,l) \\ g^{SC}_{\lambda_u,\lambda_v}(k,l) \\ g^{SS}_{\lambda_u,\lambda_v}(k,l) \end{bmatrix} = \begin{bmatrix} \cos\frac{k\pi}{N}(\lambda_u+\frac{1}{2})\cos\frac{l\pi}{N}(\lambda_v+\frac{1}{2}) \\ \cos\frac{k\pi}{N}(\lambda_u+\frac{1}{2})\sin\frac{l\pi}{N}(\lambda_v+\frac{1}{2}) \\ \sin\frac{k\pi}{N}(\lambda_u+\frac{1}{2})\cos\frac{l\pi}{N}(\lambda_v+\frac{1}{2}) \\ \sin\frac{k\pi}{N}(\lambda_u+\frac{1}{2})\sin\frac{l\pi}{N}(\lambda_v+\frac{1}{2}) \end{bmatrix}. \tag{46}$$

III. Subpel Sinusoidal Orthogonality Principles

In [18], [19], estimation of integer-pel displacements in DCT domain utilizes the sinusoidal orthogonal principles:

$$\mathrm{IDCT}\{C(k)\cos[\frac{k\pi}{N}(n+\frac{1}{2})]\} \triangleq \frac{2}{N}\sum_{k=0}^{N-1} C^2(k)\cos[\frac{k\pi}{N}(m+\frac{1}{2})]\cos[\frac{k\pi}{N}(n+\frac{1}{2})] = \delta(m-n)+\delta(m+n+1), \tag{47}$$

$$\mathrm{IDST}\{C(k)\sin[\frac{k\pi}{N}(n+\frac{1}{2})]\} \triangleq \frac{2}{N}\sum_{k=1}^{N} C^2(k)\sin[\frac{k\pi}{N}(m+\frac{1}{2})]\sin[\frac{k\pi}{N}(n+\frac{1}{2})] = \delta(m-n)-\delta(m+n+1), \tag{48}$$

where $\delta(n)$ is the discrete impulse function, and m, n are integers. This is no longer valid at the subpixel level.

In (47)-(48), we replace the integer variables m and n by the real variables u and v and define

$$\bar{L}_c(u,v) \triangleq \sum_{k=0}^{N-1} C^2(k) \cos\frac{k\pi}{N}(u+\frac{1}{2}) \cos\frac{k\pi}{N}(v+\frac{1}{2}), \tag{49}$$

$$\bar{L}_s(u,v) \triangleq \sum_{k=0}^{N-1} C^2(k) \sin\frac{k\pi}{N}(u+\frac{1}{2}) \sin\frac{k\pi}{N}(v+\frac{1}{2}). \tag{50}$$

We show in the Appendix that

$$\bar{L}_c(u,v) = -\frac{1}{2} + \frac{1}{2}[\xi(u-v) + \xi(u+v+1)], \tag{51}$$

$$\bar{L}_s(u,v) = \frac{1}{2}\sin[\pi(u+\frac{1}{2})]\sin[\pi(v+\frac{1}{2})] + \frac{1}{2}[\xi(u-v) - \xi(u+v+1)], \tag{52}$$

where

$$\xi(x) \triangleq \sum_{k=0}^{N-1} \cos(\frac{k\pi}{N}x) = \frac{1}{2}[1 - \cos\pi x + \sin\pi x \cdot \frac{\cos\frac{\pi x}{2N}}{\sin\frac{\pi x}{2N}}]. \tag{53}$$

If $\frac{(\pi x)}{(2N)}$ is so small that the second and higher order terms of $\frac{(\pi x)}{(2N)}$ can be ignored, then $\cos\frac{\pi x}{2N} \approx 1$, $\sin\frac{\pi x}{2N} \approx \frac{\pi x}{2N}$. Thus,

$$\xi(x) \approx \frac{1}{2}[1 - \cos\pi x] + N\mathrm{sinc}(x), \tag{54}$$

where $\mathrm{sinc}(x) \triangleq \sin(\pi x)/(\pi x)$. For large N, $\xi(x)$ is approximately a sinc function whose largest peak can be identified easily at $x = 0$ as depicted in Fig. 4, where $\xi(x)$ closely resembles $N \cdot \mathrm{sinc}(x)$, especially when x is small.

A closer look at (51)-(52) reveals that either $\bar{L}_c(u,v)$ or $\bar{L}_s(u,v)$ consists of ξ functions and one extra term which is not desirable. In order to obtain a pure form of sinc functions similar to (47)-(48), we define two modified functions $L_c(u,v)$ and $L_c(u,v)$ as follows:

$$L_c(u,v) \triangleq \sum_{k=0}^{N-1} \cos\frac{k\pi}{N}(u+\frac{1}{2}) \cos\frac{k\pi}{N}(v+\frac{1}{2}), \tag{55}$$

$$L_s(u,v) \triangleq \sum_{k=1}^{N-1} \sin\frac{k\pi}{N}(u+\frac{1}{2}) \sin\frac{k\pi}{N}(v+\frac{1}{2}). \tag{56}$$

Then we can show that

$$L_c(u,v) = \frac{1}{2}[\xi(u-v) + \xi(u+v+1)], \tag{57}$$

$$L_s(u,v) = \frac{1}{2}[\xi(u-v) - \xi(u+v+1)]. \tag{58}$$

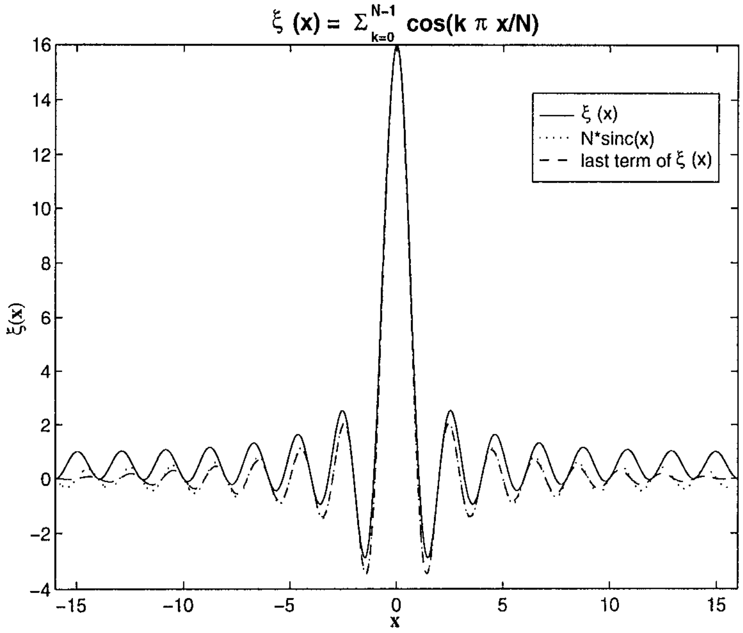
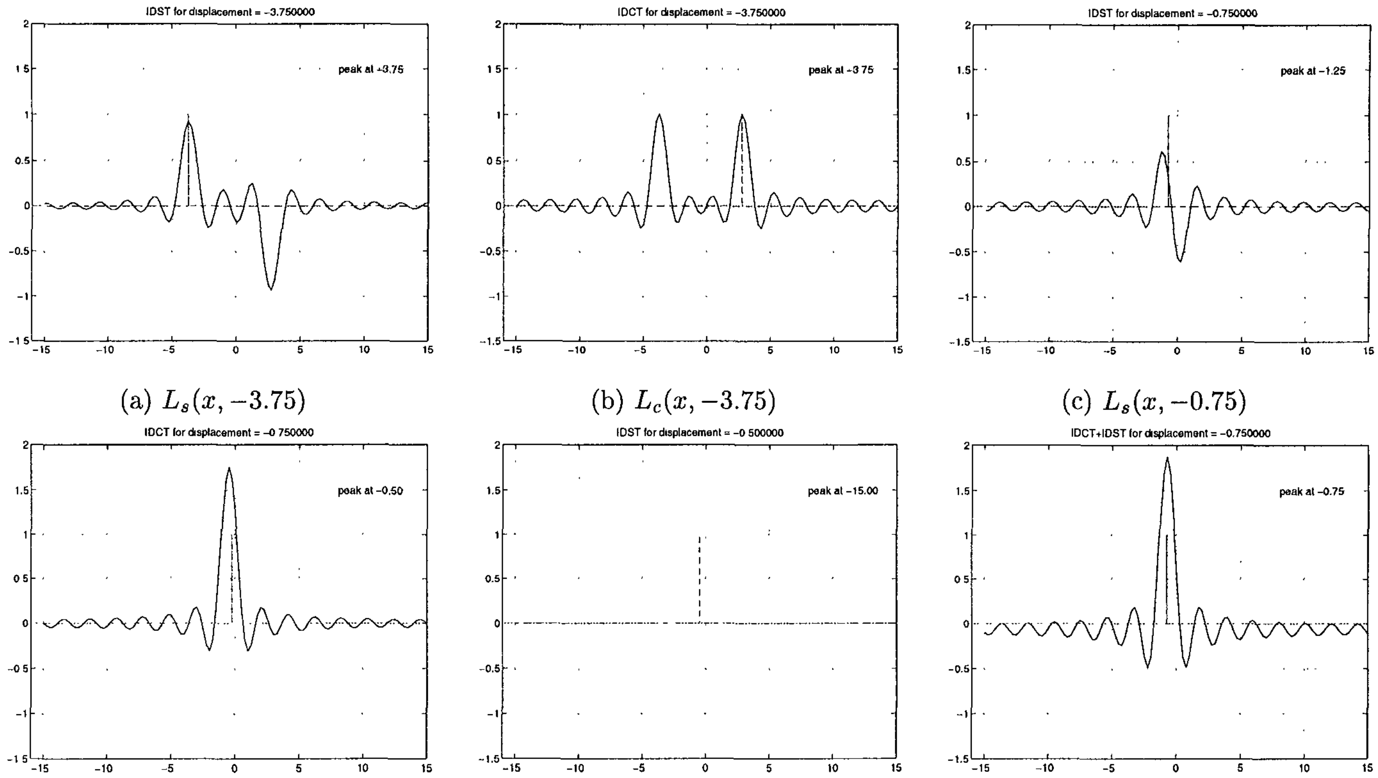


Fig. 4. Plot of $\xi(x) = \sum_{k=0}^{N-1} \cos(\frac{k\pi}{N}x)$ for $N = 16$. Observe the similarity between the curves of N*sinc(x) and the last term of ξ.

Equations (55)-(58) are the equivalent form of the sinusoidal orthogonal principles (47)-(48) at the subpixel level. The sinc functions at the right hand side of the equations are the direct result of the rectangular window inherent in the DCT transform [32]. Fig. 5 (a) and (b) illustrate $L_s(x, -3.75)$ and $L_c(x, -3.75)$ respectively where two ξ functions are interacting with each other but their peak positions clearly indicate the displacement. However, when the displacement v is small (in the neighborhood of -0.5), $\xi(u-v)$ and $\xi(u+v+1)$ move close together and addition/subtraction of $\xi(u-v)$ and $\xi(u+v+1)$ changes the shape of L_s and L_c. As a result, neither L_s nor L_c looks like two ξ functions and the peak positions of L_s and L_c are different from those of $\xi(u - v)$ and $\xi(u + v + 1)$, as demonstrated in Fig. 5 (c) and (d) respectively where the peak positions of $L_s(x, -0.75)$ and $L_c(x, -0.75)$ are -1.25 and -0.5, differing from the true displacement -0.75. In the extreme case, $\xi(u - v)$ and $\xi(u + v + 1)$ cancel out each other when the displacement is -0.5 such that $L_s(x, -0.5) \equiv 0$ as shown in Fig. 5(e).

Fortunately we can eliminate the adverse interaction of the two ξ functions by simply adding L_c and L_s together since $L_c(x, v) + L_s(x, v) = \xi(x - v)$ as depicted in Fig. 5(f) where the sum $L_c(x, -0.75) + L_s(x, -0.75)$ behaves like a sinc function and its peak position coincides with the displacement. Furthermore, due to the sharpness of this ξ function, we can accurately pinpoint the peak position under a noisy situation and in turn determine the motion estimate. This property enables us to devise flexible and scalable subpixel motion estimation algorithms in the subsequent sections.



(a) $L_s(x, -3.75)$ (b) $L_c(x, -3.75)$ (c) $L_s(x, -0.75)$

(d) $L_c(x, -0.75)$ (e) $L_s(x, -0.5)$ (f) $L_c(x, -0.75) + L_s(x, -0.75)$

Fig. 5. Illustration of sinusoidal orthogonal principles at the subpixel level for different displacements.

IV. DCT-Based Fractional-Pel Motion Estimation

In this section, we apply the subpixel sinusoidal orthogonal principles to develop an exact subpixel motion displacement scheme without the use of interpolation to estimate half-pel and quarter-pel movements for high quality video applications.

A. DCT-Based Half-Pel Motion Estimation (HDXT-ME)

From (39) in Section II, we know that the subpixel motion information is hidden, though not obvious, in the pseudo phases. To obtain subpixel motion estimates, we can directly compute the pseudo phases in (39) and then locate the peaks of the sinc functions after applying the subpixel sinusoidal orthogonal principles (55)-(58) to the pseudo phases. Alternatively, we can have better flexibility and scalability by first using the DXT-ME algorithm to get an integer-pel motion estimate and then utilizing the pseudo phase functions $f(k,l)$ and $g(k,l)$ computed in the DXT-ME algorithm as in Table I to increase estimation accuracy to half-pel, due to the fact that (39) has exactly the same form as (25). Specifically, based upon the subpixel sinusoidal orthogonal principles (55)-(58), the subpixel motion information can be extracted in the form of impulse functions with peak positions closely related to the displacement.

Sign of $\overline{DSC}$ Peak	Sign of $\overline{DCS}$ Peak	Peak Index	Direction of Motion
+	+	(λ_u, λ_v)	northeast
+	−	$(\lambda_u, -(\lambda_v + 1))$	southeast
−	+	$(-(\lambda_u + 1), \lambda_v)$	northwest
−	−	$(-(\lambda_u + 1), -(\lambda_v + 1))$	southwest

TABLE II

DETERMINATION OF DIRECTION OF MOVEMENT (λ_u, λ_v) FROM THE SIGNS OF $\overline{DSC}$ AND $\overline{DCS}$

For the sake of flexibility and modularity in design and further reduction in complexity, we adopt the second approach to devise a motion estimation scheme with arbitrary fractional pel accuracy by applying the subpixel sinusoidal orthogonal principles to the pseudo phase functions passed from the DXT-ME algorithm. The limitation of estimation accuracy will only be determined by the interaction effects of the ξ functions as explained in Section III and the slope of the ξ function at and around zero and how well the subpixel motion information is preserved in the pseudo phases after sampling.

We define $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$ as follows:

$$\overline{DCS}(u, v) \triangleq \sum_{k=0}^{N-1} \sum_{l=1}^{N-1} [\frac{f(k, l)}{C(k)C(l)}] \cos \frac{k\pi}{N}(u + \frac{1}{2}) \sin \frac{l\pi}{N}(v + \frac{1}{2}), \tag{59}$$

$$\overline{DSC}(u, v) \triangleq \sum_{k=1}^{N-1} \sum_{l=0}^{N-1} [\frac{g(k, l)}{C(k)C(l)}] \sin \frac{k\pi}{N}(u + \frac{1}{2}) \cos \frac{l\pi}{N}(v + \frac{1}{2}). \tag{60}$$

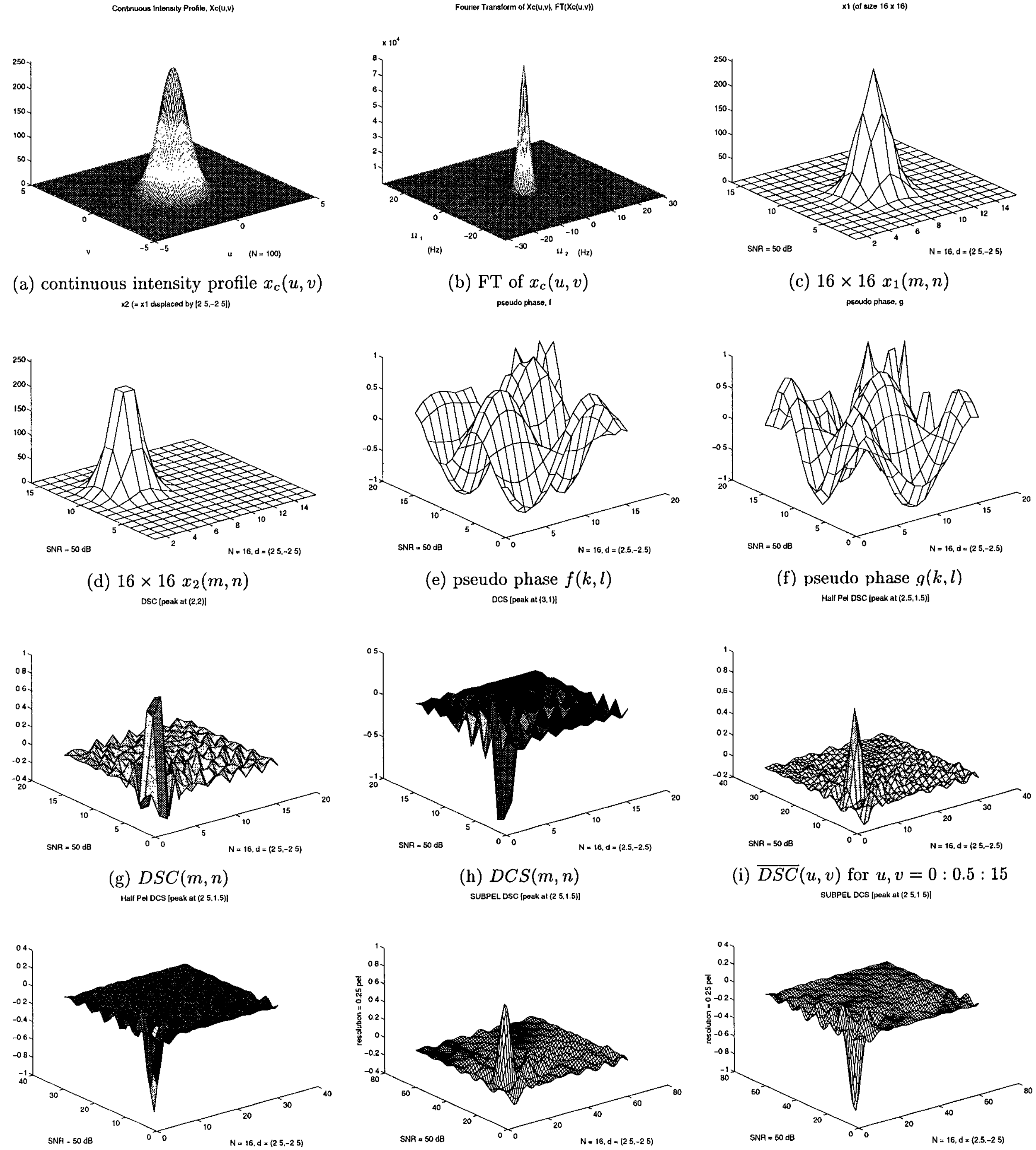
Thus, from the subpixel sinusoidal orthogonal principles (55)-(58) and the definitions of $f(k, l)$ and $g(k, l)$ in Table I, we can show that

$$\overline{DCS}(u, v) = \frac{1}{4}[\xi(u - \lambda_u) + \xi(u + \lambda_u + 1)] \cdot [\xi(v - \lambda_v) - \xi(v + \lambda_v + 1)], \tag{61}$$

$$\overline{DSC}(u, v) = \frac{1}{4}[\xi(u - \lambda_u) - \xi(u + \lambda_u + 1)] \cdot [\xi(v - \lambda_v) + \xi(v + \lambda_v + 1)]. \tag{62}$$

The rules to determine subpixel motion direction are summarized in Table II and similar to the rules in determination of integer-pel motion direction in [19].

Fig. 6 illustrates how to estimate subpixel displacements in the DCT domain. Fig. 6 (c) and (d) depict the input images $x_1(m, n)$ of size 16 × 16 (i.e. $N = 16$) and $x_2(m, n)$ displaced from $x_1(m, n)$ by $(2.5, -2.5)$ respectively at SNR = 50 dB. These two images are sampled on a rectangular grid at a sampling distance $d = 0.625$ from the continuous intensity profile $x_c(u, v) = \exp(-(u^2 + v^2))$ for $u, v \in [-5, 5]$ in Fig. 6 (a) whose Fourier transform is bandlimited as in Fig. 6 (b) to satisfy the condition



(a) continuous intensity profile $x_c(u,v)$

(b) FT of $x_c(u,v)$

(c) 16×16 $x_1(m,n)$

(d) 16×16 $x_2(m,n)$

(e) pseudo phase $f(k,l)$

(f) pseudo phase $g(k,l)$

(g) $DSC(m,n)$

(h) $DCS(m,n)$

(i) $\overline{DSC}(u,v)$ for $u,v = 0 : 0.5 : 15$

(j) $\overline{DCS}(u,v)$ for $u,v = 0 : 0.5 : 15$

(k) $\overline{DSC}(u,v)$ for $u,v = 0 : 0.25 : 15$

(l) $\overline{DCS}(u,v)$ for $u,v = 0 : 0.25 : 15$

Fig. 6. Illustration of DCT-based half-pel motion estimation algorithm (HDXT-ME)

in Theorem 1. Fig. 6 (e) and (f) are the 3-D plots of the pseudo phases $f(k,l)$ and $g(k,l)$ provided by the DXT-ME algorithm which also computes $DCS(m,n)$ and $DSC(m,n)$ as shown in Fig. 6 (g) and (h) with peaks positioned at $(3,1)$ and $(2,2)$ corresponding to the integer-pel estimated displacement vectors $(3,-2)$ and $(2,-3)$ respectively because only the first quadrant is viewed. As a matter of fact, $DCS(m,n)$ and $DSC(m,n)$ have large magnitudes at $\{(m,n); m = 2,3,\ n = 1,2\}$.

To obtain an estimate at half-pel accuracy, we calculate $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ in (59) and (60) respectively for $u,v = 0 : 0.5 : N-1$ as depicted in Fig. 6 (i) and (j) where the peaks can clearly be identified at $(2.5, 1.5)$ corresponding to the motion estimate $(2.5,-2.5)$ exactly equal to the true displacement vector even though the two input images do not look alike. Note that the notation $a : r : b$ is an abbreviation of the range $\{a + i \cdot r \text{ for } i = 0, \ldots, \lfloor \frac{b-a}{r} \rfloor\} = \{a, a+r, a+2r, \ldots, b-r, b\}$. For comparison, $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ are also plotted in Fig. 6 (k) and (l) respectively for $u,v = 0 : 0.25 : N-1 = 0, 0.25, 0.5, \ldots, N-1.25, N-1$ where smooth ripples are obvious due to the ξ functions inherent in $\overline{DCS}$ and $\overline{DSC}$ of (61)-(62) and have peaks also at $(2.5, 1.5)$.

Therefore, the DCT-based half-pel motion estimation algorithm (HDXT-ME) comprises three steps:

1. The DXT-ME algorithm estimates the integer components of the displacement as $(\hat{m}_u, \hat{m}_v)$.
2. The pseudo phase functions from the DXT-ME algorithm, $f(k,l)$ and $g(k,l)$, are used to compute $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ for $u \in \{\hat{m}_u - 0.5, \hat{m}_u, \hat{m}_u + 0.5\}$ and $v \in \{\hat{m}_v - 0.5, \hat{m}_v, \hat{m}_v + 0.5\}$ from (59) and (60) respectively.
3. Search the peak positions of $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ for the range of indices, $\overline{\Phi} = \{(u,v) : u \in \{\hat{m}_u - 0.5, \hat{m}_u, \hat{m}_u + 0.5\};\ v \in \{\hat{m}_v - 0.5, \hat{m}_v, \hat{m}_v + 0.5\}\}$, to find

$$(u_{\overline{DCS}}, v_{\overline{DCS}}) = arg \max_{u,v \in \overline{\Phi}} |\overline{DCS}(u,v)|, \tag{63}$$

$$(u_{\overline{DSC}}, v_{\overline{DSC}}) = arg \max_{u,v \in \overline{\Phi}} |\overline{DSC}(u,v)|. \tag{64}$$

These peak positions determine the estimated displacement vector $(\hat{\lambda}_u, \hat{\lambda}_v)$. However, if the absolute value of $\overline{DSC}(u,v)$ is less than a preset threshold $\epsilon_D > 0$, then $\hat{\lambda}_u = -0.5$. Likewise, if $|\overline{DCS}(u,v)| < \epsilon_D$, $\hat{\lambda}_v = -0.5$. Therefore

$$\hat{\lambda}_u = \begin{cases} u_{\overline{DSC}} = u_{\overline{DCS}}, & \text{if } |\overline{DSC}(u_{\overline{DSC}}, v_{\overline{DSC}})| > \epsilon_D, \\ -0.5, & \text{if } |\overline{DSC}(u_{\overline{DSC}}, v_{\overline{DSC}})| < \epsilon_D, \end{cases} \tag{65}$$

$$\hat{\lambda}_v = \begin{cases} v_{\overline{DCS}} = v_{\overline{DSC}}, & \text{if } |\overline{DCS}(u_{\overline{DCS}}, v_{\overline{DCS}})| > \epsilon_D, \\ -0.5, & \text{if } |\overline{DCS}(u_{\overline{DCS}}, v_{\overline{DCS}})| < \epsilon_D. \end{cases} \tag{66}$$

In Step 2, only those half-pel estimates around the integer-pel estimate $(\hat{m}_u, \hat{m}_v)$ are considered due to the fact that the DXT-ME algorithm finds the nearest integer-pel motion estimate $(\hat{m}_u, \hat{m}_v)$ from the subpixel displacement. This will significantly reduce the number of computations without evaluating all possible half-pel displacements.

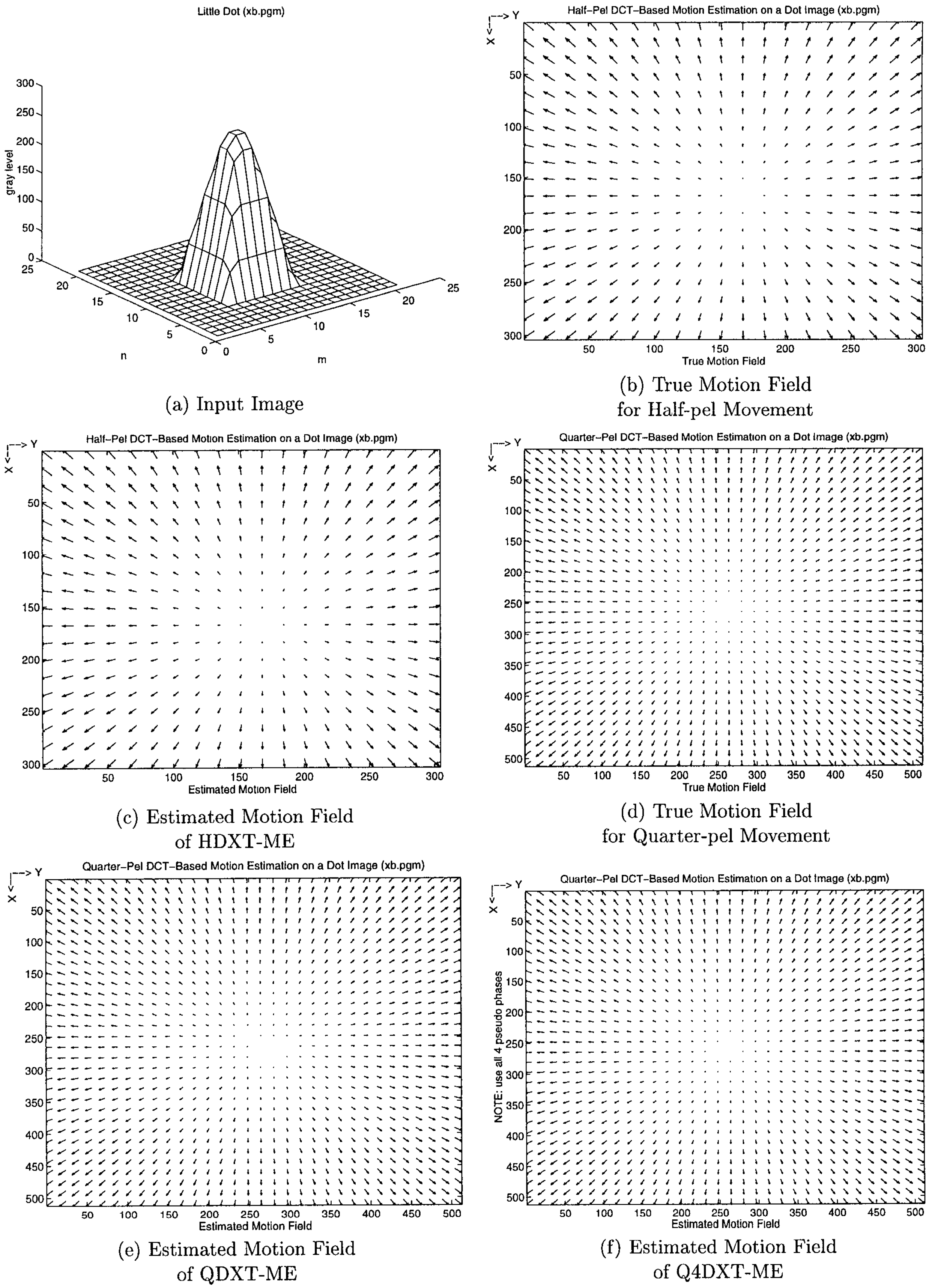
In Step 3, the use of ϵ_D deals with the case of zero pseudo phases when the displacement is -0.5. Specifically, if $\lambda_u = -0.5$, then $g^{SC}_{\lambda_u,\lambda_v}(k,l) = 0$, $\forall k,l$ which leads to $g(k,l) = 0$ and $\overline{DSC}(u,v) = 0$. However, in a noisy situation, it is very likely that $g(k,l)$ is not exactly zero and thus neither is $\overline{DSC}(u,v)$. Therefore, ϵ_D should be set very small but large enough to accommodate the noisy case. In our experiment, ϵ_D is empirically chosen to be 0.08. Similar consideration is made on $\overline{DCS}(u,v)$ for $\lambda_v = -0.5$. It is also possible that the peak positions of $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ differ in the noisy circumstances. In this case, the arbitration rule used in the DXT-ME algorithm may be applied [18], [19].

To demonstrate the accuracy of this HDXT-ME algorithm, we use a 16×16 dot image x_1 in Fig. 7 (a) as input and displace x_1 to generate the second input image x_2 according to the true motion field $\{(\lambda_u, \lambda_v) : \lambda_u, \lambda_v = -5 : 0.5 : 4\}$ shown in Fig. 7 (b) through the bilinear interpolating function specified in the MPEG standard [27] which interpolates the value $x(m+u, n+v)$ from four neighboring pixel values for m, n being integers and $u, v \in [0,1)$ in the following way:

$$\begin{aligned} x(m+u,n+v) = {} & (1-u)\cdot(1-v)\cdot x(m,n) + (1-u)\cdot v\cdot x(m,n+1) \\ & + u\cdot(1-v)\cdot x(m+1,n) + u\cdot v\cdot x(m+1,n+1). \end{aligned} \tag{67}$$

Fig. 7 (c) shows the estimated motion field by the HDXT-ME algorithm which is exactly the same as the true motion field.

Fig. 8 (a)-(c) further illustrate estimation accuracy for half-pel motion estimation schemes using peak information from $L_s(u,v)$, $L_c(u,v)$, and $L_c(u,v) + L_s(u,v)$ respectively. In Fig. 8 (a), the "+" line indicates peak positions of $L_s(u,v)$ found in the index range $\{0 : 0.5 : 15\}$ for a block size $N = 16$ with respect to different true displacement values $\{-7 : 0.5 : 7\}$. The "o" line specifies the final estimates after determination of motion directions from the peak signs of $L_s(u,v)$ according to the rules in Table II. These estimates are shown to align with the reference line $u = v$, implying their correctness. For the true displacement $= -0.5$, $L_s(-0.5, v) \equiv 0$ for all v and ϵ_D is used to decide whether the estimate should be set to -0.5. In Fig. 8 (b), $L_c(u,v)$ is used instead of $L_s(u,v)$ but $L_c(u,v)$ is always positive, inferring



(a) Input Image

(b) True Motion Field
for Half-pel Movement

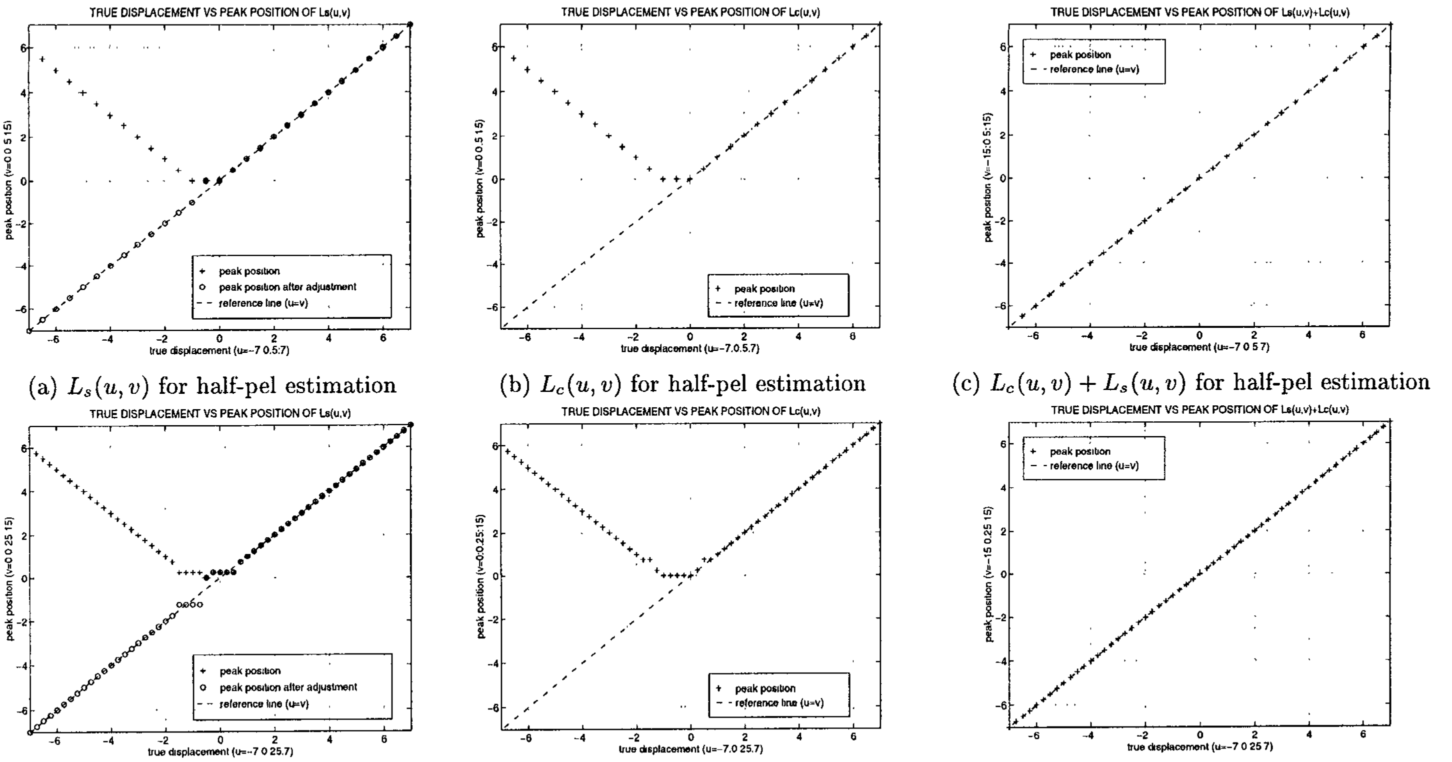
(c) Estimated Motion Field
of HDXT-ME

(d) True Motion Field
for Quarter-pel Movement

(e) Estimated Motion Field
of QDXT-ME

(f) Estimated Motion Field
of Q4DXT-ME

Fig. 7. Estimated motion fields (c)(e) of HDXT-ME and QDXT-ME by moving a dot image (a) according to the true motion fields (b)(d).



(a) $L_s(u,v)$ for half-pel estimation

(b) $L_c(u,v)$ for half-pel estimation

(c) $L_c(u,v) + L_s(u,v)$ for half-pel estimation

(d) $L_s(u,v)$ for quarter-pel estimation

(e) $L_c(u,v)$ for quarter-pel estimation

(f) $L_c(u,v) + L_s(u,v)$ for quarter-pel estimation

Fig. 8. Relation between true displacements and peak positions for half-pel and quarter-pel estimation. The signs of peak values in $L_s(u,v)$ indicate the motion directions and are used to adjust the peak positions for motion estimates.

that no peak sign can be exploited to determine motion direction. In Fig. 8 (c), $L_c(u,v) + L_s(u,v)$ provides accurate estimates without adjustment for all true displacement values but the index range must include negative indices, i. e. $[-15 : 0.5 : 15]$.

In the HDXT-ME algorithm, Step 2 involves only nine $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ values at and around $(\hat{m}_u, \hat{m}_v)$. Since $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ are variants of inverse 2D-DCT-II, the parallel and fully-pipelined 2D DCT lattice structure proposed in [7], [24], [25] can be used to compute $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ at a cost of $O(N)$ operations in N steps. Furthermore, the searching in Step 3 requires $O(N^2)$ operations for one step. Thus, the computational complexity of the HDXT-ME algorithm is $O(N^2)$ in total.

B. DCT-Based Quarter-Pel Motion Estimation (QDXT-ME and Q4DXT-ME)

In Section III, we mention that the interaction of two ξ functions in $L_c(u,v)$ and $L_s(u,v)$ from (51) and (52) disassociates the peak locations with the displacement (λ_u, λ_v) for $\lambda_u,\ \lambda_v \in [-1.5, 0.5]$. In spite of this, in the HDXT-ME algorithm, we can still accurately estimate half-pel displacements by locating the peaks of $L_s(\lambda, v)$ for true displacements $\lambda = -N+1 : 0.5 : N-1$ and indices $v = 0 : 0.5 : N-1$ if ϵ_D is introduced to deal with the case for $\lambda = -0.5$. However, at the quarter-pel level, it does cause estimation errors around $\lambda = -0.5$ as indicated in Fig. 8 (d) where the peaks of $L_s(\lambda, v)$ stay at $v = 0$ for true displacements λ varying over $[-1, 0]$. As mentioned in Section III, the sum of $L_c(\lambda, v)$ and $L_s(\lambda, v)$ is a pure ξ function and thus the adverse interaction is eliminated. As a result, the peak position of this sum can be used to predict precisely the displacement at either half-pel level or quarter-pel level as demonstrated in Fig. 8 (c) and (f) respectively. However, for two dimensional images, $\overline{DCS}$ or $\overline{DSC}$ has four ξ functions as in (61) or (62). For the DXT-ME algorithm provides two pseudo phase functions $f(k, l)$ and $g(k, l)$, only $\overline{DCS}$ and $\overline{DSC}$ are available for subpixel estimation. In this case, the sum of $\overline{DCS}$ and $\overline{DSC}$ can only annihilates two ξ functions, leaving two ξ functions as given by:

$$\overline{DCS}(u,v) + \overline{DSC}(u,v) = \frac{1}{2}[\xi(u-\lambda_u)\xi(v-\lambda_v) - \xi(u+\lambda_u+1)\xi(v+\lambda_v+1)]. \tag{68}$$

Even though this sum is not a single ξ function, the estimation error of using this sum is limited to 1/4 pixel for the worst case when true displacements are either -0.75 or -0.25.

The above discussion leads to the DCT-based quarter-pel motion estimation algorithm (QDXT-ME) as follows:

1. The DXT-ME algorithm computes the integer-pel estimate $(\hat{m}_u, \hat{m}_v)$.
2. $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ are calculated from $f(k,l)$ and $g(k,l)$ in (59) and (60) respectively for the range of indices, $\overline{\Phi} = \{(u,v) : u = \hat{m}_u - 0.75 : 0.25 : \hat{m}_u + 0.75;\ v = \hat{m}_v - 0.75 : 0.25 : \hat{m}_v + 0.75\}$.
3. Search the peak position of $D_2(u,v) \stackrel{\triangle}{=} \overline{DCS}(u,v) + \overline{DSC}(u,v)$ over $\overline{\Phi}$, i. e.

$$(u_{D2}, v_{D2}) = arg \max_{u,v \in \overline{\Phi}} |D_2(u,v)|. \tag{69}$$

The estimated displacement vector is obtained as follows:

$$(\hat{\lambda}_u, \hat{\lambda}_v) = \begin{cases} (u_{D2}, v_{D2}), & \text{if } |D_2(u_{D2}, v_{D2})| > \epsilon_D, \\ (-0.5, -0.5), & \text{if } |D_2(u_{D2}, v_{D2})| < \epsilon_D. \end{cases} \tag{70}$$

Step 3 is based on the fact that $|D_2(\lambda_u, \lambda_v)| = 0$ if and only if $(\lambda_u, \lambda_v) = -0.5$. This QDXT-ME algorithm follows the same procedure as HDXT-ME except the search region and using the sum of $\overline{DCS}$ and $\overline{DSC}$. Therefore, QDXT-ME has the same computational complexity, $O(N^2)$, as HDXT-ME.

If we modify the DXT-ME algorithm to provide the other two pseudo phase functions g^{CC} and g^{SS} in addition to f and g, we can compute $\overline{DCC}$ and $\overline{DSS}$ in the following way:

$$\overline{DCC}(u,v) \triangleq \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g^{CC}(k,l)\cos\frac{k\pi}{N}(u+\frac{1}{2})\cos\frac{l\pi}{N}(v+\frac{1}{2}), \tag{71}$$

$$\overline{DSS}(u,v) \triangleq \sum_{k=1}^{N-1}\sum_{l=1}^{N-1} g^{SS}(k,l)\sin\frac{k\pi}{N}(u+\frac{1}{2})\sin\frac{l\pi}{N}(v+\frac{1}{2}). \tag{72}$$

Then we can show that

$$D_4(u,v) \triangleq \overline{DCC}(u,v) + \overline{DCS}(u,v) + \overline{DSC}(u,v) + \overline{DSS}(u,v) = \xi(u-\lambda_u)\xi(v-\lambda_v). \tag{73}$$

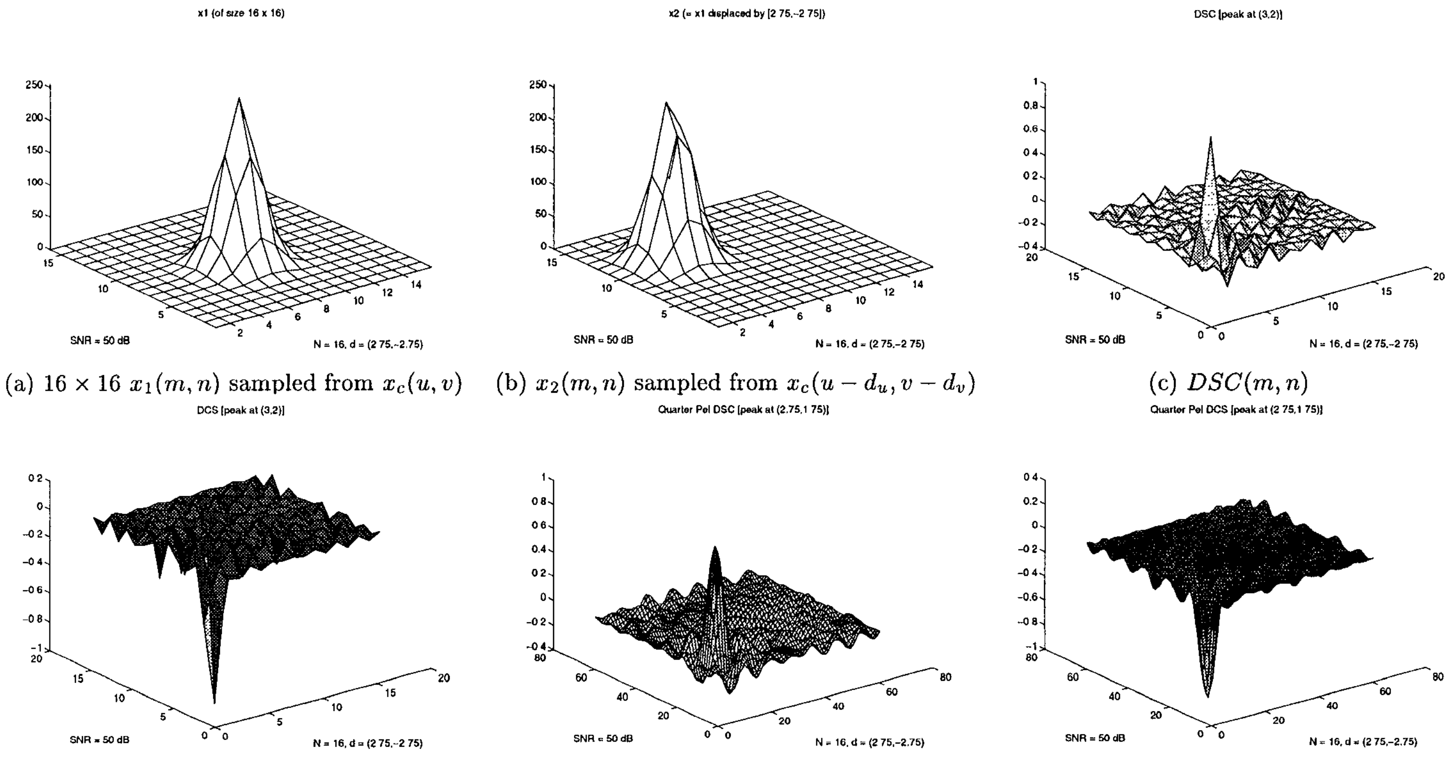
This sum[1] contains only one ξ without any negative interaction effect whose peak is sharp at (λ_u,λ_v). This leads to another quarter-pel motion estimation algorithm (Q4DXT-ME), which can estimate accurately for all displacements at the quarter-pel or even finer level.

1. Find the integer-pel estimate $(\hat{m}_u,\hat{m}_v)$ by the DXT-ME algorithm.
2. Obtain four pseudo phases g^{CC}, g^{CS}, g^{SC} and g^{SS} from the modified DXT-ME algorithm. Compute $\overline{DCS}(u,v)$, $\overline{DSC}(u,v)$, $\overline{DCC}(u,v)$, and $\overline{DSS}(u,v)$ for the range of indices, $\overline{\Phi} = \{(u,v) : u = \hat{m}_u - 0.75 : 0.25 : \hat{m}_u + 0.75;\ v = \hat{m}_v - 0.75 : 0.25 : \hat{m}_v + 0.75\}$.
3. Search the peak position of $D_4(u,v)$ over $\overline{\Phi}$: $(u_{D4},v_{D4}) = arg\max_{u,v\in\overline{\Phi}}|D_4(u,v)|$. The estimated displacement vector is then the peak position: $(\hat{\lambda}_u,\hat{\lambda}_v) = (u_{D4},v_{D4})$.

Fig. 9 shows the procedure to estimate a quarter-pel displacement with input images $x_1(m,n)$ and $x_2(m,n)$ sampled from the continuous intensity profile $x_c(u,v)$ and its shift $x_c(u-\lambda_u d, v-\lambda_v d)$ where $(\lambda_u,\lambda_v) = (2.75,-2.75)$ and $d = 0.625$ as shown in Fig. 9 (a) and (b). Fig. 9 (c) and (d) plot $DSC(m,n)$ and $DCS(m,n)$ whose peaks are both at $(3,2)$ corresponding to the integer-pel estimate $(3,-3)$. Fig. 9 (e) and (f) are the graphs of $\overline{DSC}(u,v)$ and $\overline{DCS}(u,v)$ at the quarter-pel level where the estimate is found to be $(2.75,-2.75)$.

Similar to the half-pel case, Fig. 7 (e) and (f) demonstrate the accuracy of the estimated motion fields determined by the QDXT-ME and Q4DXT-ME algorithms respectively as compared to the true motion field in Fig. 7 (d). The first input image $x_1(m,n)$ to both algorithms is a bandlimited dot image in Fig. 7 and the second input image $x_2(m,n)$ is generated by shifting $x_1(m,n)$ with respect

[1] These four functions can be generated naturally at the same time using the computing algorithms and architectures in [7], [24].



(a) 16 × 16 $x_1(m,n)$ sampled from $x_c(u,v)$ (b) $x_2(m,n)$ sampled from $x_c(u-d_u,v-d_v)$ (c) $DSC(m,n)$

(d) $DCS(m,n)$ (e) $\overline{DSC}(u,v)$ for $u,v = 0:0.25:15$ (f) $\overline{DCS}(u,v)$ for $u,v = 0:0.25:15$

Fig. 9. Illustration of DCT-based quarter-pel motion estimation algorithm (QDXT-ME)

to the true motion field in Fig. 7 (d) through the bilinear interpolation. Though not obvious in the graphs, the estimates of QDXT-ME around -0.5 have an estimation error up to a quarter pixel whereas Q4DXT-ME gives us perfect estimation.

V. Experimental Results

A set of simulations are performed on two video sequences of different characteristics: Small Flower Garden (HFG) containing fast moving scene and Miss America (MS) with slow head and shoulder movement accompanying with occasional eye and mouth opening. The performance of the DCT-based algorithms is compared with Full Search Block Matching Algorithm (BKM-ME) and its subpixel counterparts in terms of mean square error per pixel (MSE) defined as MSE $= \{\sum_{m,n}[\hat{x}(m,n) - x(m,n)]^2\}/N^2$ where $\hat{x}(m,n)$ is the reconstructed image predicted from the original image $x(m,n)$ based upon the estimated displacement vector $\hat{\lambda} = (\hat{\lambda}_u, \hat{\lambda}_v)$. For all the MSE values computed in the experiment, the bilinear interpolation in (67) is used for fair comparison to reconstruct images displaced by a fractional pixel. Furthermore, for visual comparison, all residual images, generated by subtracting the original images from the reconstructed frames predicted by various motion estimation schemes, are displayed after the saturation level is reset to 35 instead of 255 to make small pixel values of the residual images be visible. In addition, the needle maps for the estimated motion fields are superimposed on the corresponding residual images.

The integer-pel BKM-ME algorithm minimizes the MSE function of the block $\{x_1(m,n);\ m,n = 0 : 1 : N-1\}$ over the search area $\Phi = \{(m,n) : m,n = -\frac{N}{2} : 1 : N-1+\frac{N}{2}\}$ such that

$$(\hat{\lambda}_u, \hat{\lambda}_v) = arg \min_{u,v=-\frac{N}{2}:1:\frac{N}{2}} \frac{\sum_{m,n=0}^{N-1}[x_2(m,n) - x_1(m-u,n-v)]^2}{N^2}. \tag{74}$$

In the simulation, two levels of subpixel block-matching motion estimation algorithms are implemented for comparison:

1. Full Search Half-Pel Block Matching Algorithm (FHBKM-ME) — Similar to BKM-ME, FHBKM-ME searches for the displacement of minimum MSE value among all possible integer-pel and half-pel displacements in the search area Φ as such:

$$(\hat{\lambda}_u, \hat{\lambda}_v) = arg \min_{u,v=-\frac{N}{2}:\frac{1}{2}:\frac{N}{2}} \frac{\sum_{m,n=0}^{N-1}[x_2(m,n) - x_1(m-u,n-v)]^2}{N^2}. \tag{75}$$

2. Full Search Quarter-Pel Block Matching Algorithm (FQBKM-ME) — FQBKM-ME considers all integer-pel, half-pel and quarter-pel displacements within the search area Φ in finding the minimum MSE value. Precisely, the estimated displacement vector is

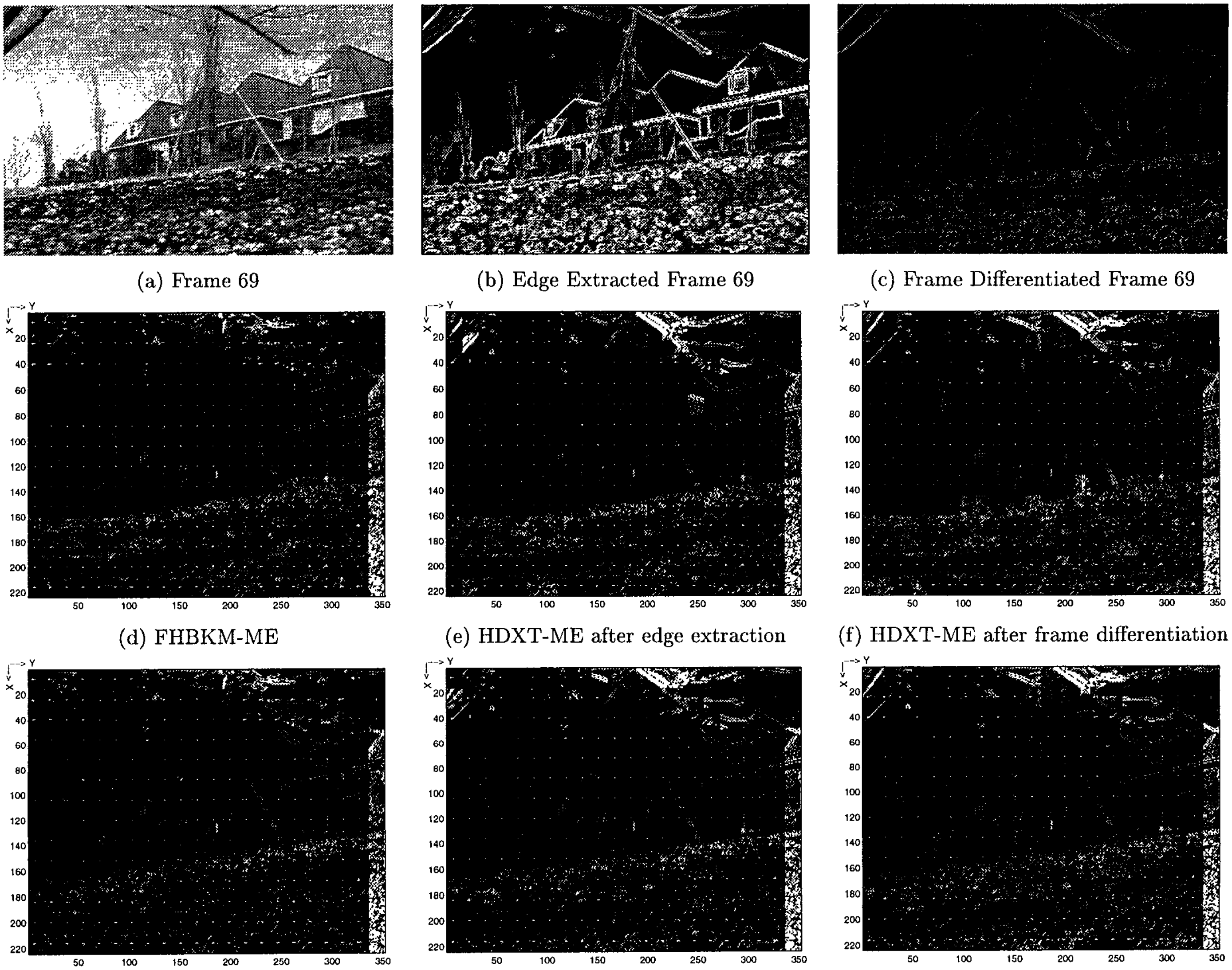
$$(\hat{\lambda}_u, \hat{\lambda}_v) = arg \min_{u,v=-\frac{N}{2}:\frac{1}{4}:\frac{N}{2}} \frac{\sum_{m,n=0}^{N-1}[x_2(m,n) - x_1(m-u,n-v)]^2}{N^2}. \tag{76}$$

Therefore, the block matching approaches should provide the optimum residuals in terms of MSE values and serve here as a benchmark of how well the DCT-based algorithms perform. It should be noted that all half and quarter pixel values for FHBKM-ME and FQBKM-ME are approximated by the bilinear interpolation. However, for the DCT-based subpixel algorithms, no interpolation is needed in finding the motion estimates. Therefore, the number of operations required by FHBKM-ME and FQBKM-ME are twice and four times as much as BKM-ME respectively whose computational complexity is $O(N^4)$ whereas the DCT-based subpixel algorithms have only marginal increase in computations over DXT-ME of which the computational complexity is $O(N^2)$. In the following simulation, simple edge extraction and frame differentiation are adopted for preprocessing input images before the DCT-based algorithms, as described in details in [18], [19]. Either preprocessing scheme adds in only $O(N^2)$ operations as overhead, keeping the total complexity remain $O(N^2)$.

The Small Flower Garden sequence has 99 frames of width 352 pixels and height 224, of which the 69^{th} Frame is shown in Fig. 10 (a). This sequence is preprocessed either by edge extractor or by frame differentiator as shown in Fig. 10 (b) and (c) respectively. It is obvious from brightness of these two subfigures that the edge extracted frame has stronger feature energy than the frame difference. As a result, it is expected that HDXT-ME and QDXT-ME perform better on edge extracted images than frame differentiated images, as clearly demonstrated by the residual images in Fig. 10 (e) versus (f) and (h) versus (i). In Fig. 10 (f) for HDXT-ME after frame difference, there are slightly more blocks of relatively high MSE than in Fig. 10 (e) for HDXT-ME after edge extraction. Compared to the residual image from FHBKM-ME in Fig. 10 (d), the residual from edge extracted HDXT-ME has similar energy distribution except at places close to the twigs at the top of the frame in Fig. 10 (e). However, at the quarter-pel level, the residuals for FQBKM-ME and QDXT-ME after either edge detection or frame difference appear similar in brightness as displayed in Fig. 10 (g)-(i). In all the residuals, bright strips at the right indicate fast left translational motion of the camera.

The simulation results on the HFG sequence are plotted in Fig. 11 where it can been seen that the DCT-based algorithms perform better on 16 × 16 blocks with 32 × 32 search areas than on 8 × 8 blocks with 16 × 16 search areas in Fig. 11 (a)-(b), in opposite to the block matching algorithms. The reason is that a larger block means more feature energy present in the block and thus it results in better estimation from the DCT-based algorithms which try to match energy patterns regardless of the shape and texture. The general trend of performance is summarized in Fig. 11 (c)-(d) where the MSE values are averaged over the whole sequence. QDXT-ME can achieve 86% reduction on average over the DIF value, the MSE value without motion compensation, as compared to 91% for FQBKM-ME for the block size 16 × 16 but the computational load for QDXT-ME is significantly lower than FQBKM-ME. Furthermore, all the graphs show that QDXT-ME has a very close performance to Q4DXT-ME in terms of MSE values.

We run simulation on Frame 60 to 90 of the MS sequence whose frame size is 352×288. The original frame 83 is shown in Fig. 12 (a) and the preprocessed frames in Fig. 12 (b)-(c) where the differentiated frame contains only very small pixel values and thus need be displayed after visualization process; otherwise, its contents will be invisible. These small DIF values indicate only slow head and shoulder motion in this sequence. The residual images for various methods in Fig. 12 (d)-(i) reveal that edge extraction is better than frame difference due to weak feature energy present in the frame differentiated



(a) Frame 69

(b) Edge Extracted Frame 69

(c) Frame Differentiated Frame 69

(d) FHBKM-ME

(e) HDXT-ME after edge extraction

(f) HDXT-ME after frame differentiation

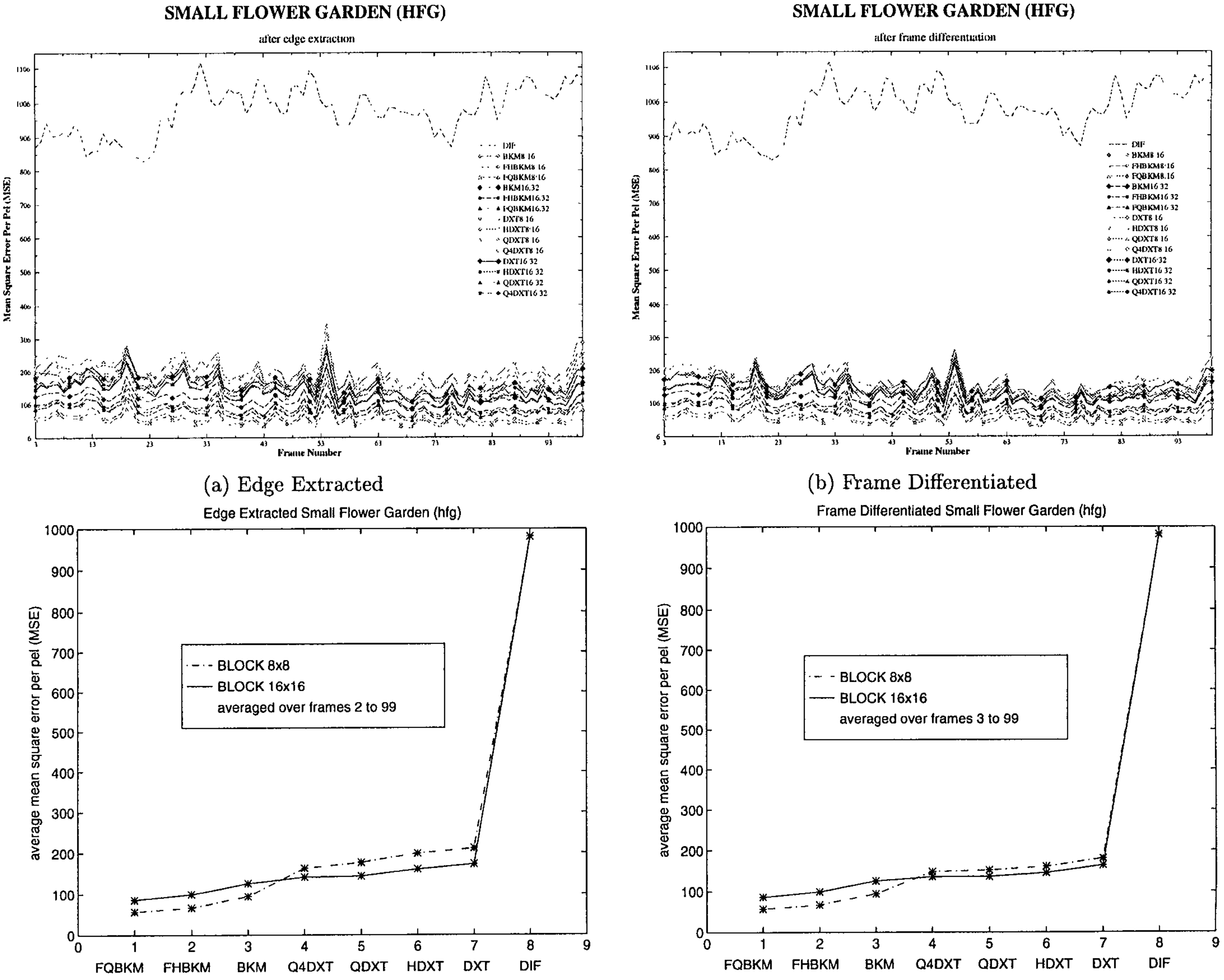
(g) FQBKM-ME

(h) QDXT-ME after edge extraction

(i) QDXT-ME after frame differentiation

Fig. 10. Comparison of different approaches on Frame 69 of Small Flower Garden sequence (HFG) for block size 16 × 16 and search size 32 × 32. Visualization is applied to (d)-(i) by setting the saturation level to 35 due to small pixel values in these residual images. The needle maps for the estimated motion fields are superimposed on the residual images.

sequence. Furthermore, there are some small patches in the clothes areas for either HDXT-ME or QDXT-ME in view of the uniform brightness in these areas removed by both preprocessing functions. This situation may be improved if a better preprocessing function is used to avoid removal of uniform areas while suppressing the aperture effect [36]. As can be seen in Fig. 13, great improvement is found for HDXT-ME, QDXT-ME and Q4DXT-ME over DXT-ME for the MS sequence, particularly after edge extraction. Due to the dominant slow motion in the sequence, QDXT-ME has 71.4% reduction over DIF in comparison with 61.9% for the integer-pel DXT-ME algorithm and Q4DXT-ME has even 72.4% gain. However, for the HFG sequence, the gain is not much but still noticeable for the subpixel



(a) Edge Extracted

(b) Frame Differentiated

(c) Preprocessed by Edge Extraction

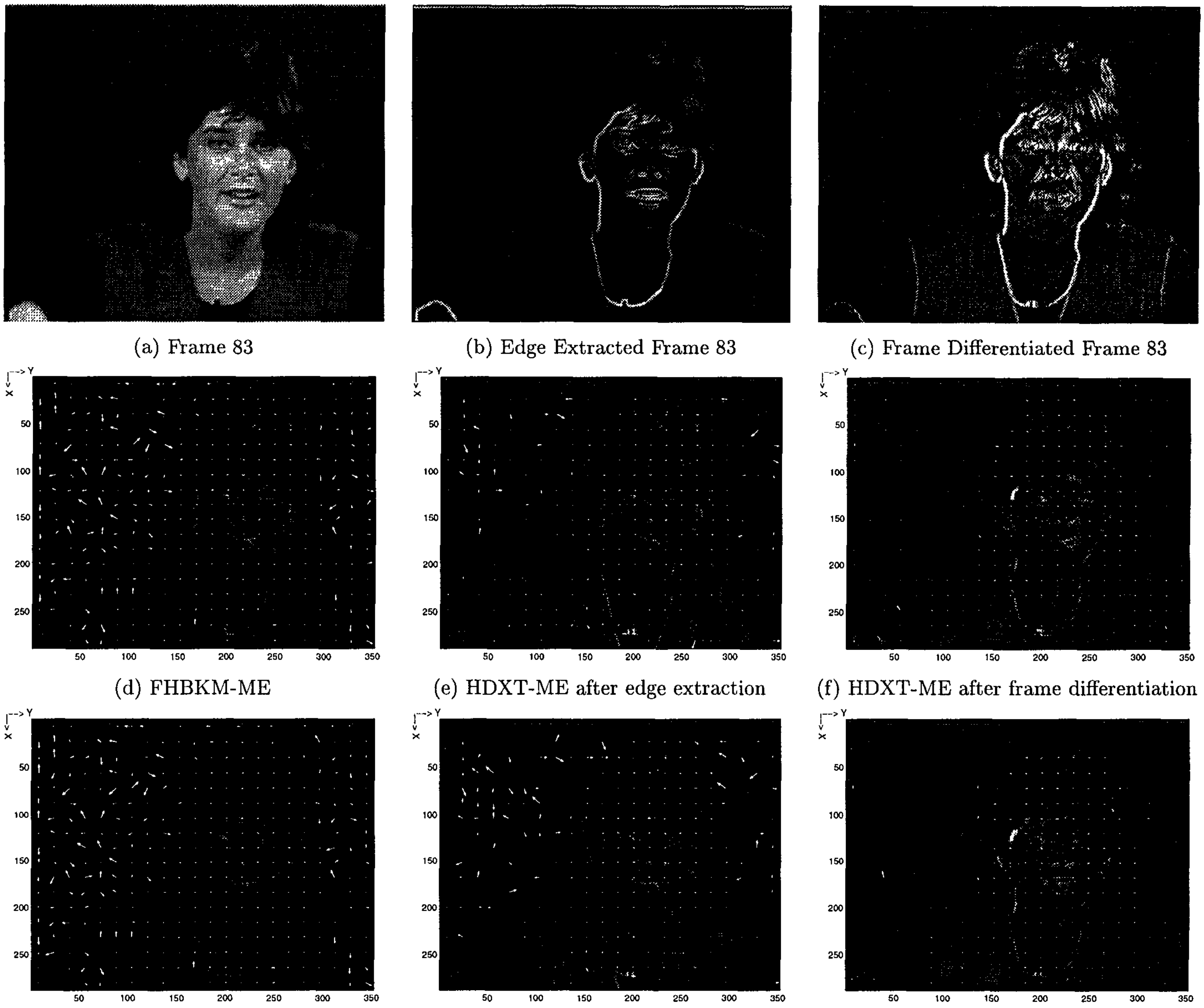
(d) Preprocessed by Frame Differentiation

Fig. 11. Simulation Results on the Small Flower Garden sequence (HFG)

algorithms over DXT-ME as in Fig. 11.

VI. Conclusion

In this paper, we develop the DCT-based subpixel motion estimation techniques based on the subpel sinusoidal orthogonal principles and preservation of subpixel motion information in DCT coefficients under the Nyquist condition. These techniques can estimate subpixel motion in DCT domain without any inter-pixel interpolation at a desired level of accuracy. Equally applicable to other areas as well, the proposed techniques are applied to video coding and result in DCT-based half-pel and quarter-pel motion estimation algorithms (HDXT-ME, QDXT-ME, Q4DXT-ME) which estimate motion with half-pel or quarter-pel accuracy without interpolation of input images. This results in significant savings in



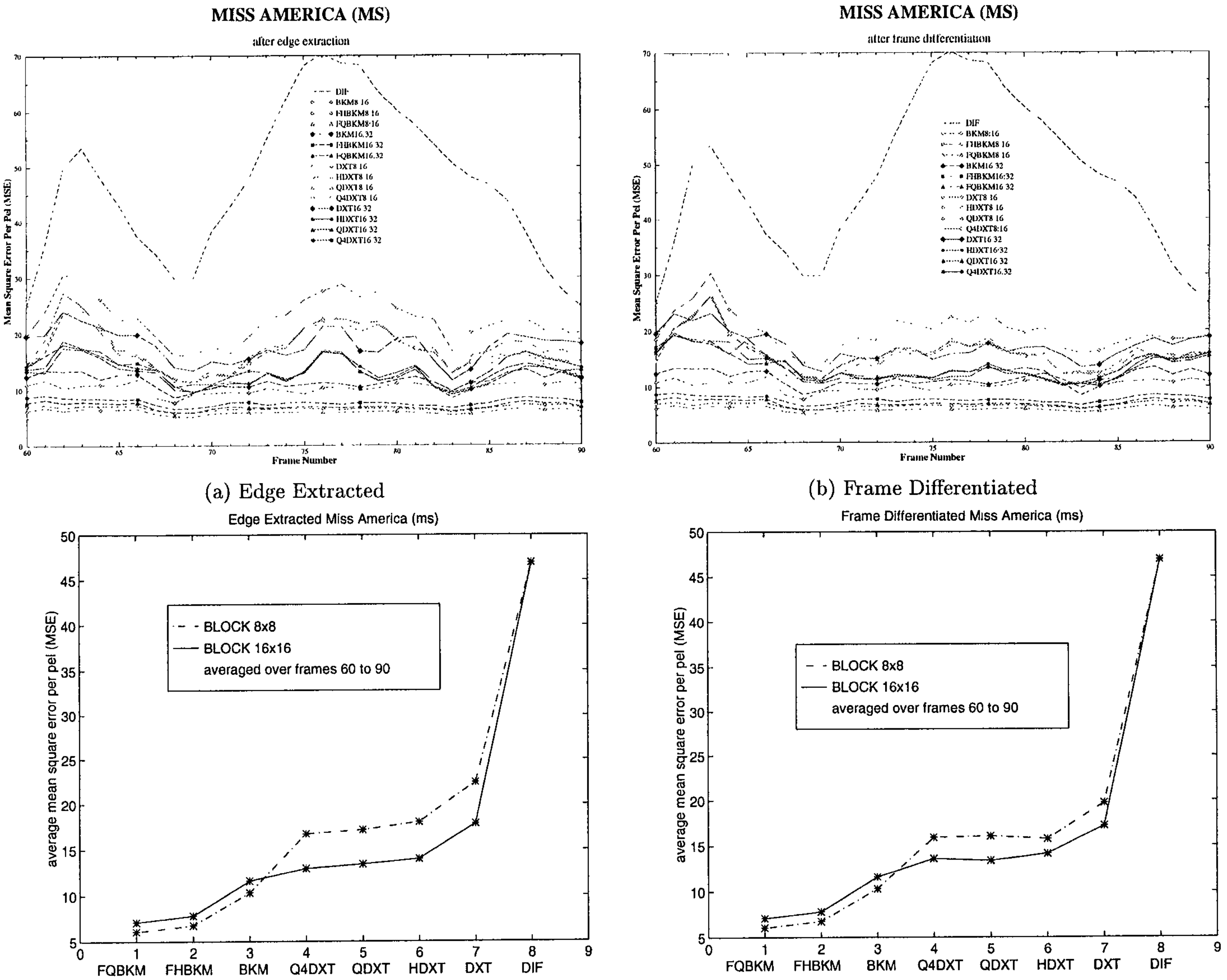
(a) Frame 83 (b) Edge Extracted Frame 83 (c) Frame Differentiated Frame 83

(d) FHBKM-ME (e) HDXT-ME after edge extraction (f) HDXT-ME after frame differentiation

(g) FQBKM-ME (h) QDXT-ME after edge extraction (i) QDXT-ME after frame differentiation

Fig. 12. Comparison of different approaches on Frame 83 of Miss America sequence (MS) for block size 16 × 16 and search size 32 × 32. Visualization is applied to (c)-(i) by setting the saturation level to 35. The needle maps for the estimated motion fields are laid over the residual images.

computational complexity for interpolation and far less data flow compared to the conventional block matching methods on interpolated images. Also, the resulted algorithms are more suitable for VLSI implementation [?, ChiuCT:92,LiuKJR:94] Furthermore, it avoids the deterioration of estimation precision caused by interpolation required in most current subpixel motion estimation schemes. In addition, the proposed DCT-based subpixel motion estimation technique and the resulted algorithms are scalable in the sense that higher estimation accuracy can be provided easily by applying the same subpel sinusoidal orthogonal principles without re-computing pseudo phases. Therefore, flexible fully DCT-based



(a) Edge Extracted

(b) Frame Differentiated

(c) Preprocessed by Edge Extraction

(d) Preprocessed by Frame Differentiation

Fig. 13. Simulation Results on the Miss America sequence (MS)

codec design is possible because the same hardware can support different levels of required accuracy. Meanwhile, the computational complexity of the DCT-based algorithms is only $O(N^2)$ compared to $O(N^4)$ for BKM-ME or its subpixel versions. Finally, HDXT-ME, QDXT-ME and Q4DXT-ME are DCT-based, enabling us to build a low-complexity and high-throughput fully DCT-based video coder.

Appendix

Equations (51)-(53) in Section III are derived as follows:

$$\bar{L}_c(u,v) \triangleq \sum_{k=0}^{N-1} C^2(k) \cos\frac{k\pi}{N}(u+\frac{1}{2}) \cos\frac{k\pi}{N}(v+\frac{1}{2}) = -\frac{1}{2} + \sum_{k=0}^{N-1} \cos\frac{k\pi}{N}(u+\frac{1}{2}) \cos\frac{k\pi}{N}(v+\frac{1}{2})$$

$$= \frac{1}{2}[-1 + \sum_{k=0}^{N-1} \cos \frac{k\pi}{N}(u - v) + \sum_{k=0}^{N-1} \cos \frac{k\pi}{N}(u + v + 1)] = -\frac{1}{2} + \frac{1}{2}[\xi(u - v) + \xi(u + v + 1)].$$

$$\begin{aligned}
\bar{L}_s(u, v) &\triangleq \sum_{k=1}^{N} C^2(k) \sin \frac{k\pi}{N}(u + \frac{1}{2}) \sin \frac{k\pi}{N}(v + \frac{1}{2}) \\
&= \frac{1}{2} \sin[\pi(u + \frac{1}{2})] \sin[\pi(v + \frac{1}{2})] + \sum_{k=0}^{N-1} \sin \frac{k\pi}{N}(u + \frac{1}{2}) \sin \frac{k\pi}{N}(v + \frac{1}{2}) \\
&= \frac{1}{2} \sin[\pi(u + \frac{1}{2})] \sin[\pi(v + \frac{1}{2})] + \frac{1}{2}[\sum_{k=0}^{N-1} \cos \frac{k\pi}{N}(u - v) - \sum_{k=0}^{N-1} \cos \frac{k\pi}{N}(u + v + 1)] \\
&= \frac{1}{2} \sin[\pi(u + \frac{1}{2})] \sin[\pi(v + \frac{1}{2})] + \frac{1}{2}[\xi(u - v) - \xi(u + v + 1)].
\end{aligned}$$

$$\begin{aligned}
\xi(x) &\triangleq \sum_{k=0}^{N-1} \cos(\frac{k\pi}{N}x) = \frac{1}{2}[\sum_{k=0}^{N-1} e^{j\frac{k\pi}{N}x} + \sum_{k=0}^{N-1} e^{-j\frac{k\pi}{N}x}] = \frac{1}{2}[\frac{1 - e^{j\pi x}}{1 - e^{j\frac{\pi x}{N}}} + \frac{1 - e^{-j\pi x}}{1 - e^{-j\frac{\pi x}{N}}}] \\
&= \frac{1}{2}[\frac{1 - \cos \pi x - \cos \frac{\pi x}{N} + \cos \frac{\pi x}{N}(N - 1)}{1 - \cos \frac{\pi x}{N}}] = \frac{1}{2}[(1 - \cos \pi x) + \frac{\sin(\pi x) \sin(\frac{\pi x}{N})}{1 - \cos(\frac{\pi x}{N})}] \\
&= \frac{1}{2}[1 - \cos \pi x + \sin \pi x \cdot \frac{\cos \frac{\pi x}{2N}}{\sin \frac{\pi x}{2N}}].
\end{aligned}$$

REFERENCES

[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images – a review", *Proceedings of the IEEE*, vol. 76, no. 8, pp. 917–935, August 1988.

[2] H. K. Aghajan, C. D. Schaper, and T. Kailath, "Machine vision techniques for subpixel estimation of critical dimensions", *Optical Engineering*, vol. 32, no. 4, pp. 828–39, April 1993.

[3] T. Akiyama, H. Aono, K. Aoki, K. W. Ler, B. Wilson, T. Araki, T. Morishige, H. Takeno, A. Sato, S. Nakatani, and T. Senoh, "MPEG2 video codec using image compression DSP", *IEEE Transactions on Consumer Electronics*, vol. 40, pp. 466–72, 1994.

[4] B. F. Alexander and K. C. Ng, "Elimination of systematic error in subpixel accuracy centroid estimation", *Optical Engineering*, vol. 30, no. 9, pp. 1320–31, September 1991.

[5] Grand Alliance, *Grand Alliance HDTV System Specification*, April 1994.

[6] D. Brinthaupt, L. Letham, V. Maheshwari, J. Othmer, R. Spiwak, B. Edwards, C. Terman, and N. Weste, "A video decoder for H.261 video teleconferencing and MPEG stored interactive video applications", in *1993 IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 1993, pp. 34–5.

[7] C. T. Chiu and K. J. R. Liu, "Real-time parallel and fully pipelined two-dimensional DCT lattice structures with applications to HDTV systems", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 25–37, March 1992.

[8] I. J. Cox, J. B. Kruskal, and D. A. Wallach, "Predicting and estimating the accuracy of a subpixel registration algorithm", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 721, August 1990.

[9] G. de Haan and W. A. C. Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching", *Signal Processing: Image Communication*, vol. 6, no. 3, pp. 229–39, June 1994.

[10] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution", *Proceedings of the IEEE*, , no. 6, pp. 858-876, June 1995.

[11] J. P. Fillard, "Subpixel accuracy location estimation from digital signals", *Optical Engineering*, vol. 31, no. 11, pp. 2465–71, November 1992.

[12] B. Girod, "Motion-compensating prediction with fractional-pel accuracy", *IEEE Trans. Communications*, vol. 41, no. 4, pp. 604, April 1993.

[13] B. Girod, "Motion compensation: Visual aspects, accuracy, and fundamental limits", in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds., chapter 5. Kluwer Academic Publishers, 1993.

[14] CCITT Recommendation H.261, *Video Codec for Audiovisual Services at p × 64 kbit/s*, CCITT, August 1990.

[15] Draft CCITT Recommendation H.263, *Line transmission of non-telephone signals: video coding for low bitrate communication*, CCITT, July 1995.

[16] S.-L. Iu, "Comparison of motion compensation using different degrees of sub-pixel accuracy for interfield/interframe hybrid coding of HDTV image sequences", in *1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA, USA, 1992, vol. 3, pp. 465–8.

[17] S. P. Kim and W. Y. Su, "Subpixel accuracy image registration by spectrum cancellation", in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, USA, 1993, vol. 5, pp. 153–6.

[18] U.-V. Koc and K. J. R. Liu, "Discrete-Cosine/Sine-Transform based motion estimation", in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Austin, Texas, November 1994, vol. 3, pp. 771–775.

[19] U.-V. Koc and K. J. R. Liu, "DCT-based motion estimation", Tech. Rep. T.R.95-1, Institute for Systems Research, University of Maryland at College Park, 1995.

[20] A. Kojima, N. Sakurai, and J. Kishigami, "Motion detection using 3D-FFT spectrum", in *ICASSP-93*, Minnesota, April 1993, IEEE, vol. V, pp. V213–V216.

[21] E. Lantz, "Subpixel signal centering and shift measurement using a recursive spectral phase algorithm", *Signal Processing*, vol. 17, no. 4, pp. 365, August 1989.

[22] X. Lee, "A fast feature matching algorithm of motion compensation for hierarchical video codec", in *Proceedings of the SPIE: Visual Communications and Image Processing '92*, Boston, MA, USA, 1992, vol. 1818, pp. 1462–74.

[23] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review", *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 589–608, September 1994.

[24] K. J. R. Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive Discrete Cosine/Sine/Hartley transforms", *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1357–1377, March 1993.

[25] K. J. R. Liu, C. T. Chiu, R. K. Kologotla, and J. F. JaJa, "Optimal unified architectures for the real-time computation of time-recursive Discrete Sinusoidal Transforms", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 2, pp. 168–180, April 1994.

[26] G. Madec, "Half pixel accuracy in block matching", in *Picture Coding Symp.*, Cambridge, MA, March 1990.

[27] CCITT Recommendation MPEG-1, *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*, ISO/IEC 11172, Geneve Switzerland, 1993.

[28] CCITT Recommendation MPEG-2, *Generic Coding of Moving Pictures and Associated Audio*, ISO/IEC 13818, Geneve Switzerland, 1994, H.262.

[29] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding", *Proceedings of the IEEE*, vol. 73, no. 4, pp. 523–548, April 1985.

[30] A. N. Netravali and J. D. Robbins, "Motion compensated television coding – part 1", *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, March 1979.

[31] A. Nosratinia and M. T. Orchard, "Discrete formulation of pel-recursive motion compensation with recursive least squares updates", in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, USA, 1993, vol. 5, pp. 229–232.

[32] A. Papoulis, *Signal Analysis*, McGraw-Hill, Inc., 1977.

[33] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architecture for video compression – a survey", *Proceedings of the IEEE*, , no. 2, pp. 220–246, February 1995.

[34] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation", *Proceedings of the IEEE*, pp. 692–702, June 1973.

[35] M. R. Shortis, T. A. Clarke, and T. Short, "A comparison of some techniques for the subpixel location of discrete target images", in *Proceedings of the SPIE: Videometrics III*, Boston, MA, USA, 1994, vol. 2350, pp. 239–50.

[36] A. Singh, *Optic Flow Computation – A Unified Perspective*, IEEE Computer Society Press, 1991.

[37] G. A. Thomas, "Television motion measurement for DATV and other applications", Tech. Rep. 11, BBC Research Department, 1987.

[38] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration", *Computer Vision, Graphics and Image Processing*, vol. 35, pp. 220–233, 1986.

[39] S.-I. Uramoto, A. Takabatake, and M. Yoshimoto, "A half-pel precision motion estimation processor for NTSC-resolution video", *IEICE Transactions on Electronics*, vol. 77, no. 12, pp. 1930, December 1994.

[40] G. A. W. West and T. A. Clarke, "A survey and examination of subpixel measurement techniques", in *Proceedings of the SPIE: Close-Range Photogrammetry Meets Machine Vision*, Zurich, Switzerland, 1990, vol. 1395, pp. 456–63.

[41] M. Ziegler, "Hierarchical motion estimation using the phase correlation method in 140 mbit/s HDTV-coding", in *Signal Processing of HDTV, II*, Turin, Italy, 1990, pp. 131–137.